

5-2015

Development and Implementation of Workshop Management System Application to Explore Combing Multiple Design Patterns

Esraa Aljahdali
St. Cloud State University

Follow this and additional works at: https://repository.stcloudstate.edu/csit_etds



Part of the [Computer Sciences Commons](#)

Recommended Citation

Aljahdali, Esraa, "Development and Implementation of Workshop Management System Application to Explore Combing Multiple Design Patterns" (2015). *Culminating Projects in Computer Science and Information Technology*. 2.
https://repository.stcloudstate.edu/csit_etds/2

This Starred Paper is brought to you for free and open access by the Department of Computer Science and Information Technology at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Computer Science and Information Technology by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

Development and Implementation of Workshop Management System

Application to Explore Combing Multiple Design Patterns

by

Esraa Aljahdali

A Starred Paper

Submitted to the Graduate Faculty

of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Science in

Computer Science

May, 2015

Starred Paper Committee:

Jayantha Herath, Chairperson

Donald Hamnes

Susantha Herath

Abstract

As systems creation are becoming more complex, we face more problems that are related to system construction rather than those problems related to analysis. Design patterns represent a clear way of solving designing problems in the context of a particular computer application. Combining design patterns is one approach that is used to allow flexibility in building applications with multiple problems.

The Workshop Management System application is built to demonstrate the idea of combining multiple design patterns. To support the idea, we show how to build the application by following the structures and methods that are provided by design patterns. Essentially, we show how these structures help developers to express a flexible system creation.

Acknowledgments

I would like to thank my committee members, Professors Jayantha Herath, Donald Hamnes, and Susantha Herath for their continuous support. Also, I would like to thank the Computer Science Department Faculty at St. Cloud State University. They provided great education that helped me to earn the skills to work on this project. Special thanks to my family for supporting me and encouraging me to do the best.

Table of Contents

	Page
LIST OF TABLE	6
LIST OF FIGURES	7
 Chapter	
1. INTRODUCTION	10
1.1 Design Patterns Definition	11
1.2 Solving Design Problems Using Design Patterns	11
2. DESIGN PATTERNS CATALOG	14
2.1 Patterns Categories Descriptions	14
2.1.1 Creational Patterns	14
2.1.2 Structural Patterns	15
2.1.3 Behavioral Patterns	15
2.2 Selecting the Right Design Pattern	15
3. COMBINING DESIGN PATTERNS	17
3.1 Workshop Management System Application	17
3.2 Descriptions of Design Patterns Used in the Application and Their Implementation	18
3.2.1 Memento Design Pattern	18
3.2.2 Strategy Design Pattern	22
3.2.3 Command Design Pattern	30
3.2.4 Observer Design Pattern	43
4. TESTING WORKSHOP MANAGEMENT SYSTEM APPLICATION	50

Chapter	Page
4.1 Application Improvement Possibilities	65
5. IMPORTANCE OF DESIGN PATTERNS	66
5.1 Conclusion	67
References	68
Appendix	70

List of Table

Table	Page
1. Design Patterns Catalog	14

List of Figures

Figure	Page
1. Memento Pattern Structure	18
2. Memento Pattern Structure Used in the Application	19
3. Memento and Origionator Class mplementation	20
4. Registration Class to Run Memento Patterns	21
5. Strategy Pattern Structure	22
6. Strategy Pattern Structure for Different Payment Strategies	23
7. Context_PaymentStrategy Implementation	24
8. PaymentStrategy Interface and Subclass Implementation	25
9. Registration Class Implementation for Payment Strategies	26
10. Strategy Pattern Structure for Different Registration Strategies	27
11. Context_RegisterStrategy, registerStrategy and TelephoneRegistration Demo	28
12. Registration Class Implementation for Registration Strategies	29
13. A Combo Box Providing a List of Registration Strategies	29
14. Command Pattern Structure	30
15. Command Pattern Structure for Different Employee Commands	32
16. Employee_Invoker and Employee_Commands Interface Implementations	33
17. AddPlace and Removeplace Classes Implementation	34
18. ChangeFee and ChangeHours Classes Implementation	35
19. Workshops Class Implementation Part 1	35
20. Implementation of Command Pattern Using Computer Graphic Workshop Example	36

Figure	Page
21. Command Pattern Structure for Different Manager Commands	37
22. Manager_Invoker and Manager_Commands Implementations	38
23. AddParticipant and RemoveParticipant Classes Implementation	39
24. addEmployee and removeEmployee Classes Implementation	40
25. ParticipantArrayList and EmployeeArrayList Implementations	41
26. ManagerPage_participant Class Implementation	42
27. ManagerPage_employee Class Implementation	43
28. Observer Pattern Structure	44
29. Observer Pattern Structure Used in the Application	45
30. Workshops Class Implementation Part 2	46
31. Reports Class Implementation	47
32. A Combo Box Providing a List of Workshop Reports	48
33. The Overall Workshop Management System Application Structure	49
34. Application Home Page	50
35. Participant Page	51
36. New Registration Page	52
37. Using Online Registration Strategy and Credit Card Payment Strategy	53
38. Using Online Registration Strategy and PayPal Payment Strategy	54
39. Registration Confirmation Detail Page	54
40. Final Window for Confirming Registration	55
41. Manager ID Page	56
42. Manage Participant Information Page	56

Figure	Page
43. Successful Participant Deletion by the Manager	57
44. Participant Canceling Registration	58
45. Participant Deleted from Participants' List in Manager Page	59
46. Successful Employee Addition Process by the Manager	60
47. Employee Addition to Employees' List in Manager Page	61
48. Employee ID Page	62
49. Employee Page for Workshops Modification	63
50. Employee Page for Displaying Workshop Reports	64
51. Reports Window Displaying Workshops Changes	64

Chapter 1: INTRODUCTION

Back to the 1970s, the architect Christopher Alexander demonstrated the first work in patterns area [1]. He studied several structures that were designed and used to solve the same problem. Later, many authors were influenced by Alexander ideas about patterns. Accordingly, they used some of Alexander's ideas to develop a set of patterns called Design Patterns for developing well-designed user interfaces [1]. A design pattern is best known for essential solution that is used to solve a problem that reappears in a specific set of situations [2]. As we build more complex computer systems, the problems that occur with system construction grow in complexity. The solution for such problems using design patterns is by combining different patterns within the system. For this reason, it was decided to create a workshop management system application that is combining multiple design patterns into one program structure to perform some operations, such as addition commands, substitution commands, restoring information, displaying reports with system changes, and selecting the preferred strategies of payment and registration. These operations are performed by implementing Strategy, Observer, Memento, and Command design patterns.

In this paper, Chapter 1 begins the paper introduction by defining design patterns. Further, the chapter discusses the design problems that design patterns can solve.

Chapter 2 discusses design patterns catalog and the definitions of Creational, Structural, and Behavioral design patterns.

Chapter 3 discusses the meaning of combining several design patterns and also discusses the reason for using patterns combination. In addition, this chapter discusses Workshop Management System application. It discusses the design patterns used and the implementations of these patterns in the application.

Chapter 4 focuses on how the application works by providing snapshots. Moreover, it explains the improvement possibilities in the Workshop Management System application.

Chapter 5 represents the benefits of using design patterns and provides the conclusion that summarizes the work and findings of Chapter 3.

1.1 Design Patterns Definition

Design patterns are descriptions of collaborating classes and objects that are used to solve a general design problem in a particular context [3]. They solve recurring software design problems by using best practices that are proven methods to build high quality software [2], [3]. Using design patterns in the process of software design and development makes full use of existing successful development experience [1]. According to Christopher Alexander, “Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” [4]. Alexander also says that a pattern is “a recurring solution to a common problem in a given context and system of forces” [4]. Each design pattern usually includes multiple participants, such as operations, classes, and attributes. Each participant has a name that represents its job. When an application is created, the names of the design patterns and their participants applied in the application usually reflect the application domain [5].

1.2 Solving Design Problems Using Design Patterns

There are several ways to solve some of the design issues using design patterns. The following list describes how design patterns handle these problems:

- Determining object granularity

Object-oriented programs are made up of objects. Object granularity is the size and number of objects. Usually, objects differ in size and number. Some objects represent an entire system and others represent subsystems. Design patterns help finding object granularity by specifying what types of objects are required for implementing the pattern [6].

- Specifying object interfaces

In object-oriented systems, objects are accessible via their interfaces. Design patterns describe interfaces by realizing their key elements and data type that should be handled. Moreover, design patterns help identify the relationship between interfaces. For example, patterns can allow some objects with different implementation to share the same interface [6].

- Specifying object implementations

A pattern describes the plan for object implementation by specifying abstract classes, concrete classes, class inheritance, interface inheritance, and class diagrams that represent relationships between classes. Class inheritance means reusing the parent class functionality to extend the child class functionality. On the other hand, interface inheritance means exchanging the use of an object with another where both objects implements the same interface, and accordingly they belong to the same type [6].

- Putting reuse mechanisms to work

Many people are able to realize the meaning of inheritance, objects, interfaces, and classes, but the actual challenge is being able to apply these concepts to build an easy, reusable system. In object-oriented systems reusability is accomplished via object class inheritance and composition. From this point, pattern definitions help specifying which reusability techniques apply to the pattern [6].

- Designing for change

When creating a system, we must put in consideration how the system could be changed later and accepts modifications. If the system design does not take changes into account, this may force redesigning the whole system instead of adding a few modifications. Design patterns provide solutions for this problem by specifying several parts included in the system creation, such as dependencies on software, hardware, platform, object representation or implementation, and other operations. Also, design patterns specify extending functionality by creating sub-classes, tight coupling, and algorithmic dependencies in order to make changes to classes convenient and flexible [6].

Chapter 2: DESIGN PATTERNS CATALOG

Design patterns catalog was first represented by Gamma et al. It is a collection of related patterns, where these patterns are distributed over three categories [7]. Table 1 represents design patterns catalog and shows three common categories that represent patterns. These categories are Creational, Structural, and Behavioral design patterns.

Table 1

Design Patterns Catalog [7]

Purpose	Creational	Structural	Behavioral
Design Pattern examples	<ul style="list-style-type: none"> • Abstract Factory • Builder • Prototype • Singleton 	<ul style="list-style-type: none"> • Adapter • Bridge • Composite • Decorator • Facade • Flyweight • Proxy 	<ul style="list-style-type: none"> • Chain of Responsibility • Command • Iterator • Mediator • Observer • State • Strategy • Visitor • Memento

2.1 Patterns Categories Descriptions

2.1.1 Creational Patterns

Creational patterns are design patterns that are concerned with mechanisms of object creation, and try to create objects that are suitable to the situation [1]. The object creation main form could result in design problems or added complexity to the design [1]. Creational design patterns take care of this problem by controlling the creation of the object. They allow the details of the classes that are instantiated to be encapsulated and hidden from the client class. Hence, the client will not be able to know how, and when these classes are created [1].

2.1.2 Structural Patterns

Structural Design Patterns are design patterns that make the design easier by representing a simple way to understand relationships between entities [6]. They use composition to unite objects and classes to put them into a larger structure [4]. These patterns show how to put several pieces of a system together in a flexible way. Moreover, if one of the parts changes, Structural patterns assure that there is no change to the entire system structure. They also show how to reorganize pieces that do not fit into pieces that do fit [4].

2.1.3 Behavioral Patterns

Behavioral design patterns are design patterns that focus on how objects interact with each other. They focus on how different classes and objects communicate with each other in order to perform operations [4]. Furthermore, they represent how the steps of an operation are assigned among different objects. By doing that, these patterns carry out the communication in a flexible way and reduce duplication and complexity of code. Moreover, Behavioral design patterns define the mechanism for selecting different algorithms by different objects at runtime [1].

2.2 Selecting the Right Design Pattern

There are many design patterns that are available. The question is, which pattern is a better solution to solve a design problem? To answer this question and find which pattern is better, there are several methods that help to approach the best patterns to a specific design problem:

- One approach is to understand how patterns relate to each other [6]. Studying the relationships between patterns can help to find the right pattern or more than one pattern that are suitable as solutions to a specific problem [6].
- In addition, we can check patterns that have the same purpose. For example, if we have a problem that needs an object to be created, we should look for patterns that come under the Creational design patterns [8].
- Another approach is to find a reason for redesign [6]. For example, if we need to expand the functionality of a design by subclassing, we can look at the design patterns that fit this kind of problem, such as Strategy design pattern [6].
- Check patterns names that represent the role of the patterns and then, find the pattern that is related to the problem; so, it can be applied to the design of the solution [8].

Chapter 3: COMBINING DESIGN PATTERNS

With the increasing size of the project, the problems that need to be solved grow in complexity. Therefore, it is natural to combine design patterns to solve complex problems. Combining design patterns means that patterns can be used together in a single design. Also, it means that two or more patterns are used together in a design that can be applied over many problems [4]. The importance of developing and combining different patterns is that there are many problems that require use of multiple patterns together within a system, which provides a significant amount of flexibility and extensibility in a software system [4]. For example, an entire organization system rationally has more than one problem to solve, even in the smallest organizations. Therefore, the system requires more than one design pattern to solve the system problems.

3.1 Workshop Management System Application

Workshop Management System is an application that combines Memento, Strategy, Command, and Observer patterns. The application is created using Java language. The system contains a list of workshops choices for the participant to choose from. Also, the program includes a number of management system functionalities, such as allowing the manager to display a list of participants that are enrolled in the available workshops. The manager is able to add or remove participants from workshops. Furthermore, the manager is able to add or remove employees from the system. In addition, the application allows the employees to modify the information of workshops, such as modifying the number of available places in the workshops, and modifying the fee and period of the workshops. For participants, the application allows them to make new registrations in the available workshops and cancel their registrations if they prefer to do this operation.

3.2 Descriptions of Design Patterns Used in the Application and Their Implementations

3.2.1 Memento Design Pattern

Memento design pattern comes under the Behavioral patterns. It is used to save the history of an object so it can be retrieved later if it is needed [9]. One of the best examples to demonstrate the Memento design pattern is undo operation represented in any word processing application. Sometimes, the user wants to create an object that existed previously, for example, trying to return to an earlier version of work that the user previously suspended. This can be done by using undo operation that is implemented using the Memento pattern [10]. Next, Figure 1 displays Memento pattern structure.

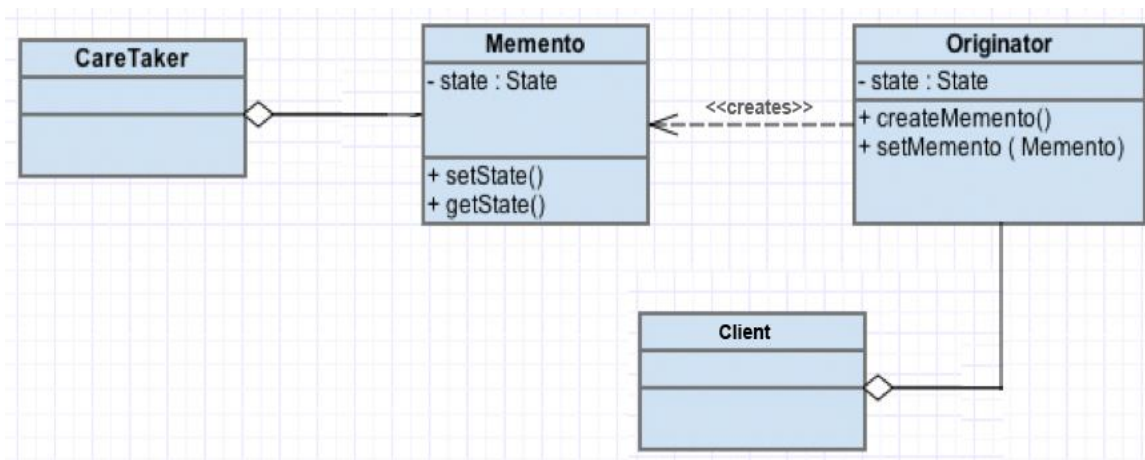


Figure 1

Memento Pattern Structure [1], [11]

As in Figure 1, the structure of Memento pattern has different classes:

- Memento: it is a private inner class of the Originator class. This means that the Originator is allowed to access Memento methods. However, outside classes cannot access these methods. As a result, the state information can be transferred

between the Memento and the Originator classes, and only the Originator object can access the state data stored in the Memento class [9].

- Originator: is the object whose state we want to save. It creates a new instance of Memento class that contains its state and passes it to the Caretaker class [9].
- Caretaker: the Caretaker class requests the Originator object to save its state. It is responsible for storing and restoring the Originator object's state through a Memento object. The Caretaker class does not know how to interact with a Memento object, but it stores the Memento object in the file system or keeps it for a time and passes it back to the Originator object [9].
- Client: Holds a copy of the Originator [11].

3.2.1.1 Implementing Memento Pattern in the Application

Memento pattern is used to save participant's data and use undo operation to restore it to the previous saved data. Memento, Originator, and Registration are the classes used to implement the Memento pattern in the application. Figure 2 represents Memento pattern structure that is used in the application.

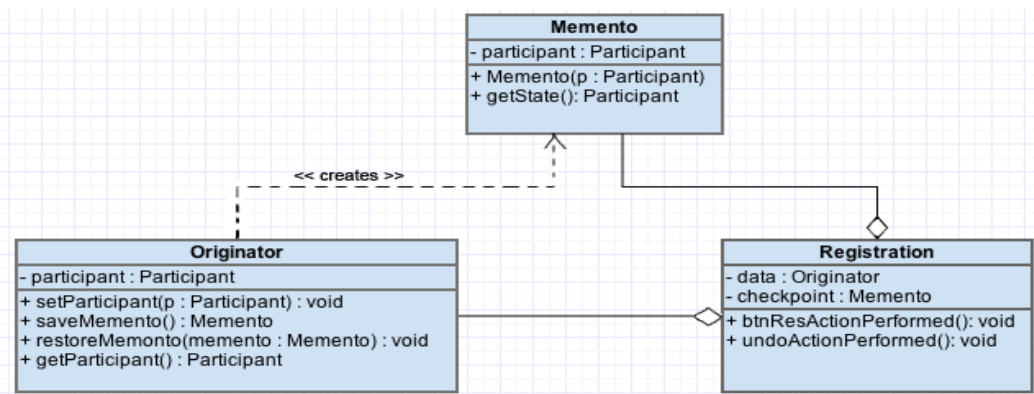


Figure 2

Memento Pattern Structure Used in the Application

Memento class is responsible for storing participant's data. The Originator class knows how to save participant's data by creating a memento object that holds a snapshot of the Originator current participant's object. Registration is the client class that is responsible for running Memento pattern in the application. Figure 3 shows the implementation of Memento and Originator classes.

```
public class Memento {  
    private Participant participant;  
    public Memento(Participant p) {  
        participant = p;  
    }  
    public Participant getState(){  
        return participant;  
    }  
}  
  
public class Originator  
{  
    private Participant participant;  
  
    public void setParticipant(Participant p){  
        participant = p;  
    }  
    public Memento saveMemento(){  
        return new Memento(participant);  
    }  
    public void restoreMemento(Memento memento) {  
        participant = memento.getState();  
    }  
    public Participant getParticipant(){  
        return participant;  
    }  
}
```

Figure 3

Memento and Originator Classes Implementation

Memento class uses `getState()` method to return the saved data of the Participant object. Originator class uses `saveMemento()` method to create a memento object that captures the current participant's data. On the other hand, `restoreMemento()` method uses the created memento object to restore participant's data that is saved previously. Method

`getParticipant()` is used to return participant's data by invoking different methods that are implemented in the Participant class. To run Memento pattern, Figure 4 shows the part in Registration class that is responsible for implementing the pattern.

```
static Originator data = new Originator();
static Memento checkpoint;

private void btnResActionPerformed() {
    Participant c = new Participant(txtFname.getText(), txtLname1.getText());

    data.setParticipant(c);
    checkpoint = data.saveMemento();
}

private void UndoActionPerformed() {

    data.restoreMemento(checkpoint);

    // to display the data in text fields.
    txtFname.setText(data.getParticipant().getFame());
    txtLname1.setText(data.getParticipant().getLame());
}
```

Figure 4

Registration Class to Run Memento Pattern

Registration class creates Originator, Memento and Participant objects. Participant object is created using the data that is retrieved from the user. Originator object sets the data by calling `setParticipant(c)` method and passes the Participant object as a parameter. Then, the Originator object creates a memento by calling `saveMemento()` method that will save the Participant object.

To restore the data of the participant, the user clicks on “undo” button that is provided in the application. Then, the method `UndoActionPerformed()` that is presented in the code will restore the participant's data. To restore the data, the Originator object calls `restoreMemento(checkpoint)` method and passes the Memento object as a parameter.

To get a specific data of the participant, we use the Originator object to call method `getParticipant()` that can access the data of the participant. From Figure 4, `data.getParticipant().getFame()` is an example of how to get the first name of the participant.

3.2.2 Strategy Design Pattern

Strategy design pattern comes under the Behavioral patterns. In this pattern, different strategies that implement a common operation or algorithm are encapsulated in separate classes [10]. These classes implement the same interface that is represented to the client. The operation method that is represented in the interface defines the input and output of the strategy. Each individual class implements the operation in a different way [10]. Since the individual classes implement the same interface, they are interchangeable at the runtime of the system [10]. Thus, if we have several algorithms for a specific task, Strategy pattern allows the client to choose the actual implementation to be used at runtime. Figure 5 represents the structure of Strategy pattern.

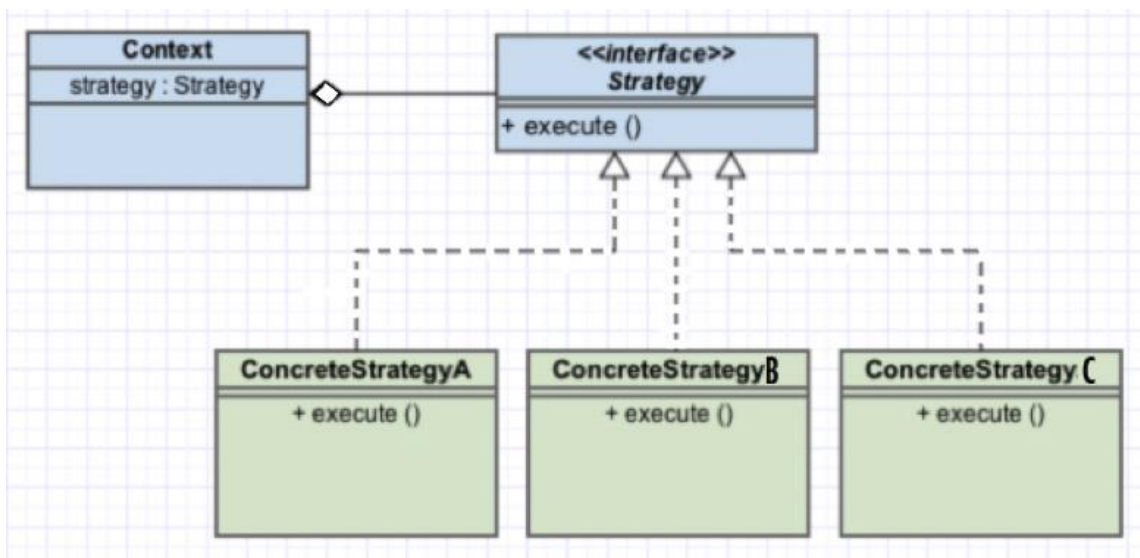


Figure 5**Strategy Pattern Structure [4], [11]**

As in Figure 5, there are different classes that represent Strategy pattern structure:

- Strategy: it is an interface that defines family of algorithms [10].
- Context: it takes care of the data during the communication with the client. It also realizes the different algorithms implemented where each algorithm is considered as a strategy [10].
- ConcreteStrategy: provides an implementation of the Strategy interface. Each ConcreteStrategy class can implement the interface in a different way [10].

3.2.2.1 Implementing Strategy Pattern in the Application

Strategy pattern is implemented twice in the application. First, Strategy pattern is used to allow the participants to choose the way of payment they prefer. They can choose cash, Visa or PayPal to do the payment. Figure 6 represents the structure of Strategy pattern using different payment strategies.

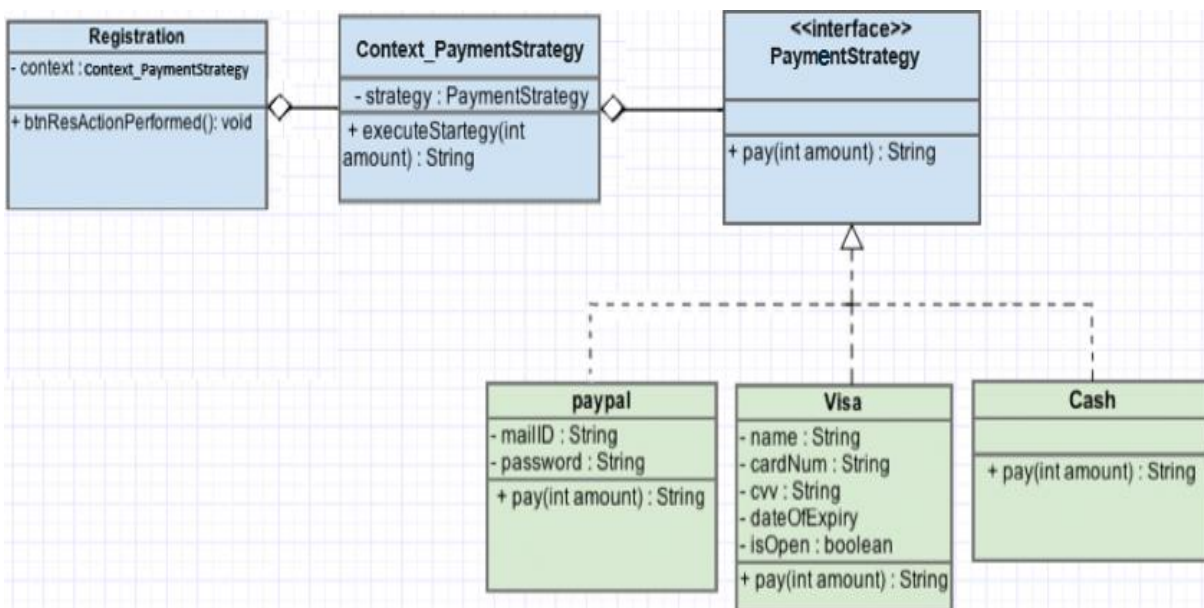


Figure 6**Strategy Pattern Structure for Different Payment Strategies**

As in Figure 6, Registration class is used to run Strategy pattern in the application. PaymentStrategy is an interface that is implemented by Visa, PayPal, and Cash concrete strategy subclasses. All three subclasses implement `pay()` method. Moreover, Context_paymentStrategy uses PaymentStrategy interface to call method `pay()` that is defined by the three concrete strategy subclasses. Figure 7 shows the implementation of Context_paymentStrategy.

```
public class Context_PaymentStrategy {
    private PaymentStrategy strategy;
    public Context_PaymentStrategy(PaymentStrategy strategy1)
    {
        this.strategy = strategy1;
    }
    public String executeStrategy(int amount) {
        return strategy.pay(amount);
    }
}
```

Figure 7**Context_PaymentStrategy Implementation**

Visa, PayPal, and Cash subclasses use `pay()` method to return a string that verifies the amount required for the workshop enrollment and the way of payment to the participant. The only difference between Visa, Cash, and PayPal subclasses is that each one of them implements `pay()` method differently and requires different variables according to the way of payment. For example, if the participant uses PayPal to do the payment, he needs to enter his PayPal account details, while using cash means the participant needs to pay the workshop's fee in office. Therefore, defining variables, such as the account number or

password is not needed in Cash subclass, while it is needed for the PayPal subclass to get the account details from the user. The implementation of PaymentStrategy interface and Visa, PayPal, and Cash subclasses are shown in Figure 8.

```
public interface PaymentStrategy {
    String pay(int amount);
}

public class Cash implements PaymentStrategy {
    public String pay(int amount) {
        String string = amount + "$ fee should be paid within 3 days in office";
        return string;
    }
}

public class Visa implements PaymentStrategy{
    private String name, cardNumber, cvv, dateOfExpiry;

    public Visa(String nm, String ccNum, String cvv, String expiryDate) {
        this.name=nm;
        this.cardNumber=ccNum;
        this.cvv=cvv;
        this.dateOfExpiry=expiryDate;
    }

    public String pay(int amount){
        String string = amount + " will be paid with visa card";
        return string;
    }
}

public class Paypal implements PaymentStrategy{
    private String emailId;
    private String password;

    public Paypal(String email, String pwd) {
        this.emailId = email;
        this.password = pwd;
    }

    public String pay(int amount){
        String string = amount + "$ fee will be paid using Paypal";
        return string;
    }
}
```

Figure 8

PaymentStrategy Interface and Subclasses Implementation

To run Strategy pattern and allow the use of different payment strategies, Registration class creates a Context_PaymentStrategy object that use PaymentStrategy interface to call pay() method. For example, if cash payment is selected, Context_PaymentStrategy object calls its constructor Context_PaymentStrategy() and passes Cash() subclass object as a parameter to identify that the payment strategy chosen is cash. After that, Context_PaymentStrategy object calls its method executeStrategy() to execute pay() method identified in Cash strategy subclass.

The executeStrategy() method will return a string of the cash payment details. See Figure 9 for Registration class implementation for different payment strategies.

```
Context_PaymentStrategy context;

private void btnResActionPerformed(java.awt.event.ActionEvent evt)
{
    if (cash.isSelected())
    {
        context = new Context_PaymentStrategy(new Cash());
        String amountMsg = (context.executeStrategy(Integer.parseInt(txtPrice.getText())));

    else if (visa.isSelected())
    {
        context = new Context_PaymentStrategy(new Visa(txtvisa2.getText(),
        txtvisa.getText(), txtvisa3.getText(), txtvisal.getText()));

        String amountMsg = (context.executeStrategy(Integer.parseInt(txtPrice.getText())));

    else if (Paypal.isSelected())
    {
        context = new Context_PaymentStrategy(new Paypal(txtpaypal.getText(),
        jPasswordField2.getText()));
        String amountMsg = (context.executeStrategy(Integer.parseInt(txtPrice.getText())));
    }
}
```

Figure 9

Registration Class Implementation for Payment Strategies

The second use of Strategy pattern in the application is to allow the participants to choose the way of registration they prefer. The registration in a workshop can be done by

office, online, telephone or email. Figure 10 represents the structure of Strategy pattern using different registration strategies.

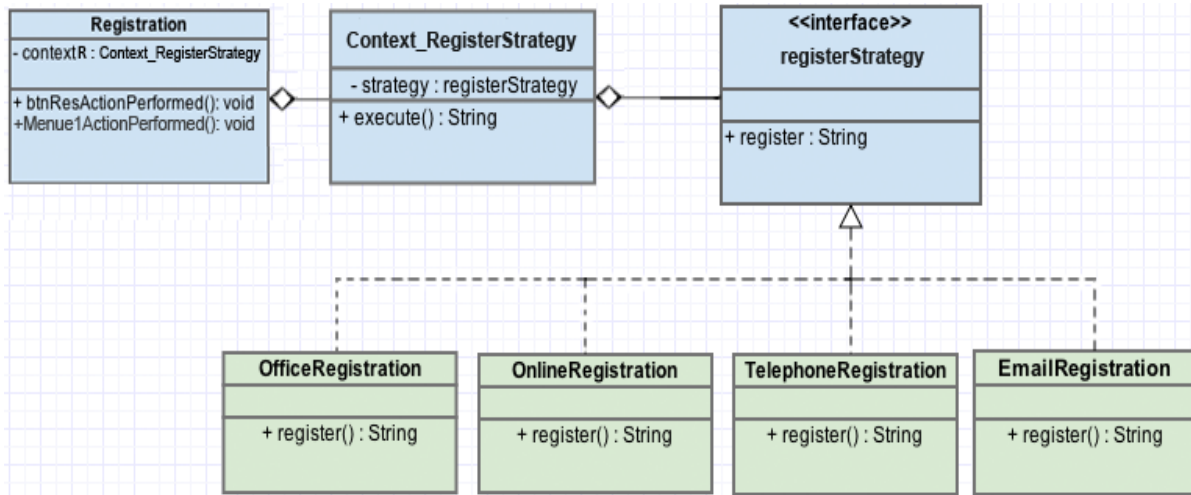


Figure 10

Strategy Pattern Structure for Different Registration Strategies

The interface `registerStrategy` is implemented by `OnlineRegistration`, `EmailRegistration`, `OfficeRegistration`, and `TelephoneRegistration` ConcreteStrategy subclasses. All subclasses implement `register()` method that returns a string displaying the information of the registration strategy. In addition, `Context_RegisterStrategy` uses `registerStrategy` interface to call method `register()` that is defined by all subclasses. `Registration` class is used to run Strategy pattern in the application. Figure 11 shows the implementation of `Context_RegisterStrategy`, `registerStrategy` interface, and `TelephoneRegistration` subclass.

```

class Context_RegisterStrategy {
    private registerStrategy registerS;

    public Context_RegisterStrategy(registerStrategy registerS)
    {
        this.registerS = registerS;
    }
    public String execute() {
        return registerS.register();
    }
}

interface registerStrategy {
    String register();
}

class TelephoneRegistration implements registerStrategy {

    public String register()
    {
        return "Dear participant, please contact Registration \n"
            + " Services (320) 567-5674 \n"
            + " (M-F 8am - 4:30pm, Sat. & Sun. 8am - 2pm) \n"
            + " or CS Workshop office 320-432-1234. " ;
    }
}

```

Figure 11

Context_RegisterStrategy, registerStrategy and TelephoneRegistration Demo

To run the second Strategy pattern used in the application, Registration class creates a Context_RegisterStrategy object that use registerStrategy interface to call register() method. Then, Context_RegisterStrategy object calls its constructor Context_RegisterStrategy() and passes the registration subclass object as a parameter. Subsequently, Context_RegisterStrategy object calls its method execute() to execute register() method identified in the subclass. See Figure 12 for Registration class implementation for different registration strategies.

```

private void MenuelActionPerformed() {
    if (Menuel.getSelectedIndex() == 0) {
        contextR = new Context_RegisterStrategy(new EmailRegistration());
        message2.setText(contextR.execute());
    }
    else if (Menuel.getSelectedIndex() == 1) {
        contextR = new Context_RegisterStrategy(new TelephoneRegistration());
        message2.setText(contextR.execute());
    }
    else if (Menuel.getSelectedIndex() == 2) {
        contextR = new Context_RegisterStrategy(new OfficeRegistration());
        message2.setText(contextR.execute());
    }
    else if (Menuel.getSelectedIndex() == 3) {
        contextR = new Context_RegisterStrategy(new OnlineRegistration());
        message2.setText(contextR.execute());
    }
}

```

Figure 12

Registration Class Implementation for Registration Strategies

The choice of registration can be retrieved using a menu listed in a Combo Box provided by Java Graphical User Interface. Each item in the menu list is identified by its index number. For example, in Figure 12 “Menuel.getSelectedIndex() == 0” means the first registration strategy in the list is chosen. See Figure 13 for the Combo Box used for different registration strategy choices.

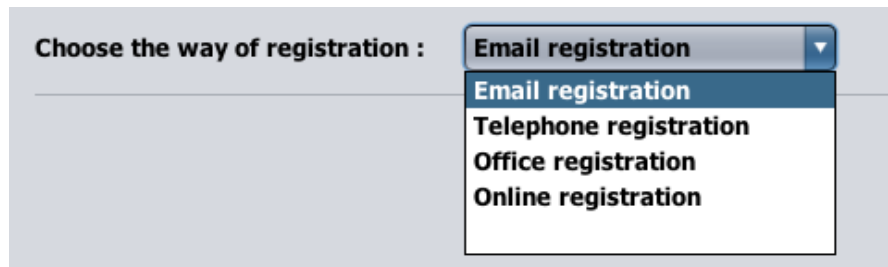


Figure 13

A Combo Box Providing a List of Registration Strategies

3.2.3 Command Design Pattern

The Command design pattern comes under the Behavioral patterns. It encloses a request for an operation inside an object. It also decouples the client that makes the request for the operation from the object that can do the operation [11]. This means the client makes requests without having any idea about the actual action that will be performed [1]. This allows the code developer to modify the action without affecting the client program [1]. Menus represent a basic example of the need for using command pattern. For example, when a program developer creates a menu in an application, he has to configure the menu with words that express the use of actions, such as Add and Remove. In addition, the program developer has to configure the menu with methods that can take care of the actions: so, the user can get a response when one of the command menu choices is clicked [10], [11]. See Figure 14 for Command pattern structure.

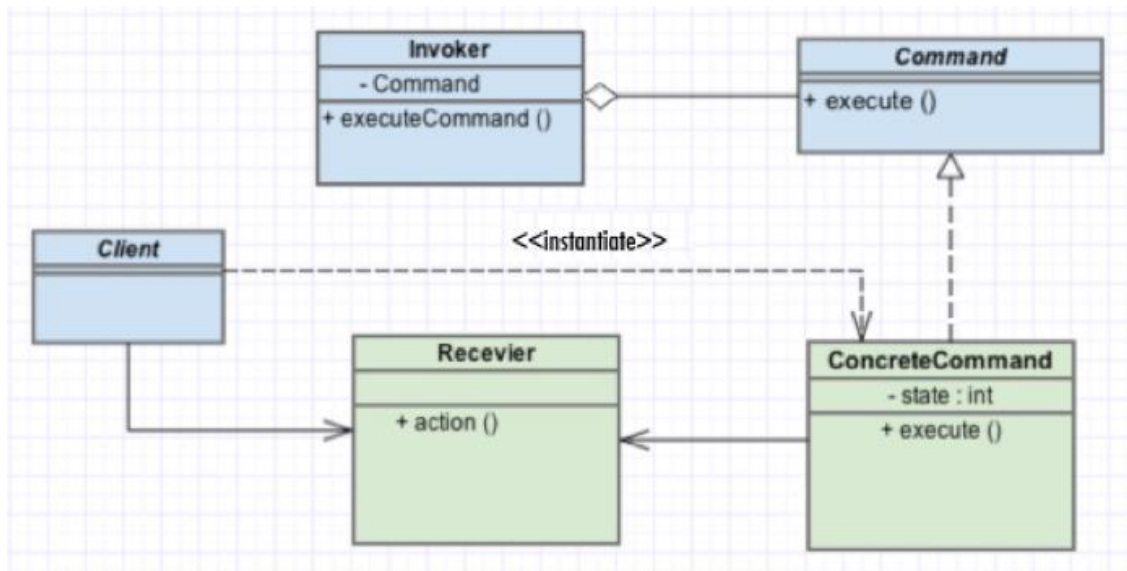


Figure 14

Command Pattern Structure [1], [4]

As in the Figure 14, the structure of Command pattern has different classes:

- **Command:** declares an interface for executing a specific action. This provides the base class for all objects [4].
- **ConcreteCommand:** it creates a link between the receiver and the action by invoking the corresponding action on the receiver [4].
- **Client:** creates an instance of the ConcreteCommand and then associates it with the receiver [4].
- **Invoker:** the client object controls the invoker object. The invoker asks the command to make an action that is requested by the client [4].
- **Receiver:** is the object that realizes how to perform the action. Its methods are executed when one or more commands are invoked [4].

3.2.3.1 Implementing Command Pattern in the Application

Command pattern is used twice in the application. First, it is used to allow the employee to make changes in the workshop registration, for example, adding or removing participants. Moreover, the employee is allowed to modify the workshop registration fee, and period. Figure 15 shows the structure of Command pattern used to implement different employee commands.

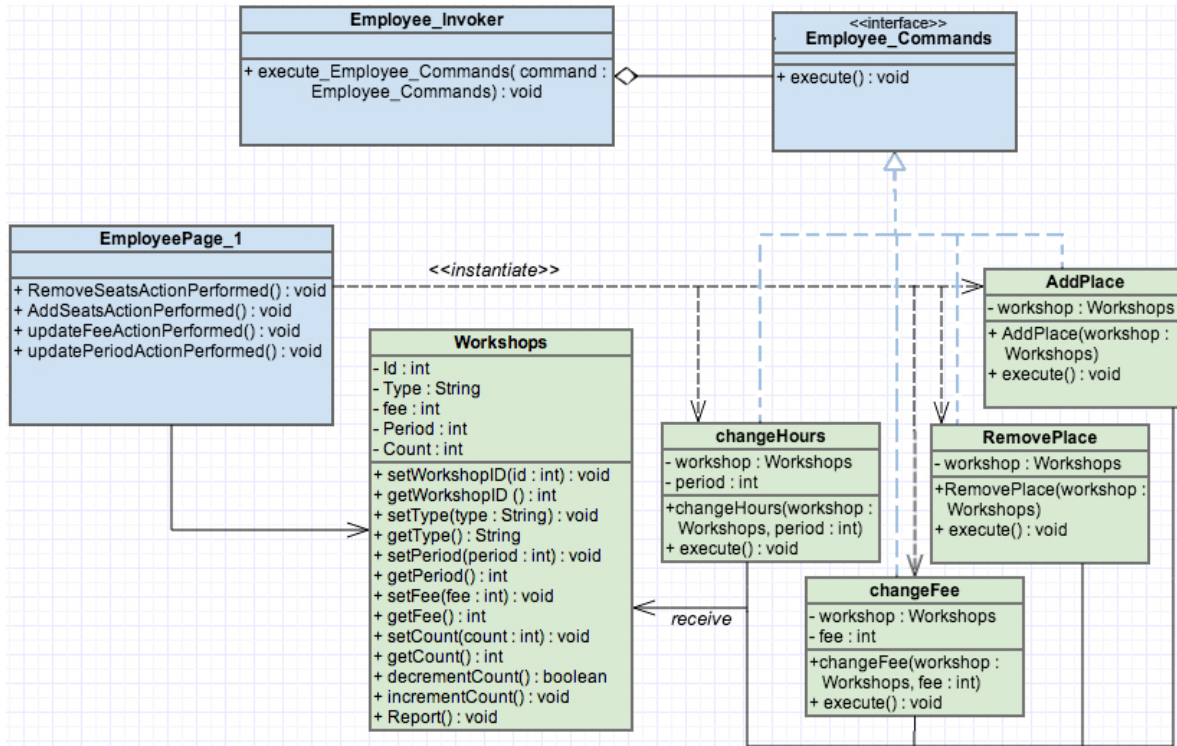


Figure 15

Command Pattern Structure for Different Employee Commands

Employee_Commands is an interface that is implemented by different concrete command classes, such as AddPlace, RemovePlace, changeFee, and changeHours.

The Employee_Invoker acts as an invoker object and uses Command pattern to identify which object will execute a command based on the type of the command. Workshops class is responsible for performing the action related to the command request using the appropriate method implemented in the class. And to demonstrate Command pattern, we use EmployeePage_1 class. The following Figure 16 represents the implementation of Employee_Invoker class and Employee_Commands interface.

```
class Employee_Invoker {  
    public Employee_Invoker() {}  
  
    void execute_Employee_Commands(Employee_Commands command) {  
        command.execute();  
    }  
}  
  
public interface Employee_Commands {  
    public void execute();  
}
```

Figure 16

Employee_Invoker and Employee_Commands Interface Implementations

All concrete command classes implement method `execute()`. The differences between the classes are the number of parameters passed to the classes' constructors and the methods that are called while executing the command. For example, both `AddPlace` and `RemovePlace` classes attach `Workshops` class object to their constructors. `AddPlace` class uses the attached object to call `incrementCount()` method, which satisfies the request need for increasing the number of places in the chosen workshop, while `RemovePlace` class uses the attached object to call `decrementCount()` method to decrease the number of places. See Figure 17 for `AddPlace` and `Removeplace` classes implementation.

```
public class AddPlace implements Employee_Commands {
    private Workshops workshop;
    public AddPlace(Workshops workshop) {
        this.workshop = workshop;
    }
    public void execute() {
        workshop.incrementCount();
    }
}

public class RemovePlace implements Employee_Commands {
    private Workshops workshop;
    public RemovePlace(Workshops workshop) {
        this.workshop = workshop;
    }
    public void execute() {
        workshop.decrementCount();
    }
}

}
```

Figure 17

AddPlace and Removeplace Classes Implementation

Both changeFee and changeHours classes attach two parameters to their constructors. For example, changeFee class attaches Workshops class object and an integer that represents the new fee entered by the employee. The object is used to call `setFee()` method, while the integer is attached to the method to set the new fee to the chosen workshop. See Figure 18 for changeFee and changeHours classes implementation.

```

public class changeFee implements Employee_Commands {
    private Workshops workshop;
    int fee;

    public changeFee(Workshops workshop, int fee) {
        this.workshop = workshop;
        this.fee = fee;
    }

    public void execute(){
        workshop.setFee(fee);
    }
}

public class changeHours implements Employee_Commands {
    private Workshops workshop;
    int period;

    public changeHours(Workshops workshop, int period) {
        this.workshop = workshop;
        this.period = period;
    }
    public void execute()
    {
        workshop.setPeriod(period);
    }
}

```

Figure 18

ChangeFee and ChangeHours Classes Implementation

Once Workshops class receives requests from different concrete command classes, it executes the actions associated with the requests. Figure 19 shows part of the implementation of Workshops class. The complete implementation of Workshops class is presented in the Appendix.

```

public class Workshops extends Observable{
    int ID, Fee, Period, Count;

    public void setPeriod(int period) {
        Period = period;
    }

    public void setFee(int fee){
        Fee = fee;
    }
    public void setCount(int count){
        Count=count;
    }
    public boolean decrementCount(){
        if(this.Count==0) {
            return false;
        }
        else {
            Count--;
        }
        return true;
    }
    public void incrementCount(){
        Count++;
    }
}

```

Figure 19

Workshops Class Implementation Part 1

Running the first use of command pattern is accomplished using EmployeePage_1 class. For example, the employee requests for adding place to a workshop. In EmployeePage_1 class, we create the workshop object that receives the request, such as Computer Graphics workshop object. Then, attach the object to AddPlace command method. Following this, we create the invoker object and attach AddPlace command object to perform the action. AddPlace concrete command will be in charge of the requested command and send its result to Workshops class, which is the receiver. Figure 20 shows the implementation of the second use of command pattern. The figure represents Computer Graphic workshop example and how to add place, remove place, change the fee, and change the period of the workshop.

```
Employee_Invoker invoker = new Employee_Invoker();
static Workshops ComputerGraphics = new Workshops(1,"Computer Graphics",200,10, 21);

private void RemoveSeatsActionPerformed() {
    Employee_Commands remove;
    if(Menuue.getSelectedIndex()==0){
        remove = new RemovePlace (ComputerGraphics);
        invoker.execute_Employee_Commands(remove);
    }
}

private void AddSeatsActionPerformed() {
    Employee_Commands add;
    if(Menuue.getSelectedIndex()== 0) {
        add = new AddPlace (ComputerGraphics);
        invoker.execute_Employee_Commands(add);
    }
}

private void updateFeeActionPerformed() {
    Employee_Commands chgFee;
    if(Menuue.getSelectedIndex()==0) {
        chgFee = new changeFee(ComputerGraphics, Integer.parseInt(enterFee.getText()));
        invoker.execute_Employee_Commands(chgFee);
    }
}

private void updatePeriodActionPerformed() {
    Employee_Commands chgPeriod;
    if(Menuue.getSelectedIndex() == 0) {
        chgPeriod = new changeHours(ComputerGraphics, Integer.parseInt(enterPeriod.getText()));
        invoker.execute_Employee_Commands(chgPeriod);
    }
}
}
```

Figure 20

Implementation of Command Pattern Using Computer Graphic Workshop Example

The second use of Command pattern in the application is to allow the manager to make changes to the employee's status, for example, adding or removing employees. If an employee is added, he will be able to access the system and make modification to the available workshops. Similarly, if an employee is removed, he will not be able to access and use the system functionalities provided for the employees. Besides changing the employee's status, the manager is allowed to add or remove participants from workshops. The Command Pattern Structure for different manager commands is represented in Figure 21.

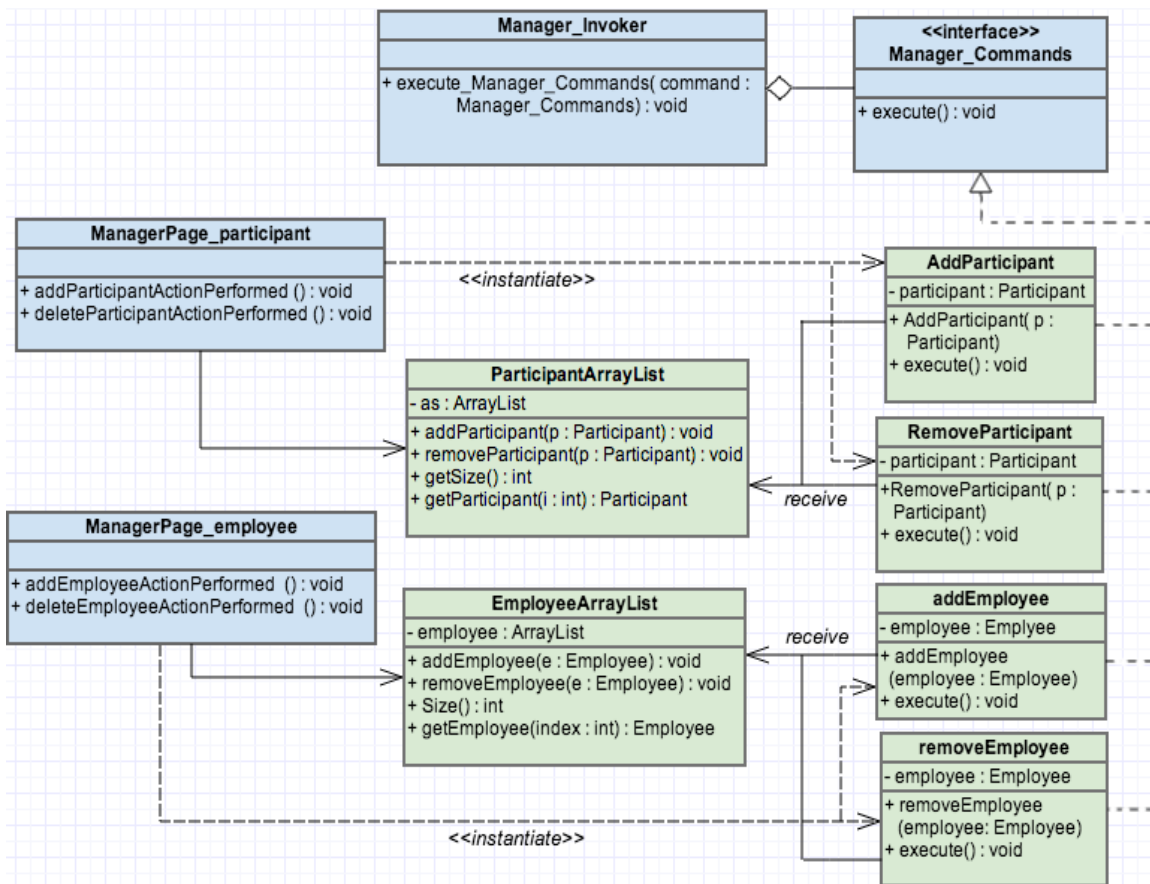


Figure 21

Command Pattern Structure for Different Manager Commands

Manager_Commands is an interface that is implemented by different concrete command classes, such as AddParticipant, RemoveParticipant, addEmployee, and removeEmployee. The Manager_Invoker acts as an invoker object and uses Command Pattern to identify which object will execute a command based on the type of the command. ParticipantArrayList class is responsible for performing the action related to the command based on the command type, AddParticipant or RemoveParticipant. Likewise, EmployeeArrayList class is responsible for performing the action related to the command based on the command type, addEmployee, or removeEmployee. In this part, two classes are used to run Command pattern, ManagerPage_participant and ManagerPage_employee. The reason for using two classes is to have different graphical user interfaces, one for managing the participants and the other one for managing the employees. The following Figure 22 shows the implementation of Manager_Invoker and Manager_Commands interface.

```
public class Manager_Invoker {  
    void execute_Manager_Commands(Manager_Commands command) {  
        command.execute();  
    }  
}  
  
public interface Manager_Commands {  
    public void execute();  
}
```

Figure 22

Manager_Invoker and Manager_Commands Implementations

All concrete command classes implement method `execute()`. The only difference between the classes is the object type identified in each class. Both AddParticipant and RemoveParticipant classes make modification to participants; therefore, the object's type identified in these classes is Participant. The two concrete classes implement `execute()`

method by modifying the participants' arraylist identified in ManagerPage_participant class. For example, the arraylist object calls its method `addParticipant()` implemented in ParticipantArrayList class to add a new participant and passes the Participant object as a parameter. ParticipantArrayList class then, requests the addition of a new participant. See Figure 23 for AddParticipant and RemoveParticipant classes implementation.

```
public class AddParticipant implements Manager_Commands{
    private Participant p;

    public AddParticipant(Participant participant) {
        this.p = participant;
    }

    public void execute() {
        ManagerPage_participant.as.addParticipant(p);
    }
}

public class RemoveParticipant implements Manager_Commands {
    private Participant participant;

    public RemoveParticipant(Participant participant){
        this.participant = participant;
    }

    public void execute(){
        ManagerPage_participant.as.removeParticipant(participant);
    }
}
```

Figure 23

AddParticipant and RemoveParticipant Classes Implementation

Similarly, `addEmployee` and `removeEmployee` classes make modification to employees; thus, the object's type identified in these classes is `Employee`. The `execute()` method implemented in these classes modifies the employees' arraylist identified in ManagerPage_employee class. For example, the arraylist object can add or remove employees

by calling its methods `addEmployee()` or `removeEmployee()` implemented in `EmployeeArrayList` class. `EmployeeArrayList` class then, performs the requested action. The implementation details of `addEmployee` and `removeEmployee` classes are shown in Figure 24.

```
public class addEmployee implements Manager_Commands {
    private Employee employee;

    public addEmployee(Employee employee) {
        this.employee = employee;
    }
    public void execute(){
        EmployeeIDPage.EID.addEmployee(employee);
    }
}

public class removeEmployee implements Manager_Commands {
    private Employee employee;

    public removeEmployee(Employee employee) {
        this.employee = employee;
    }
    public void execute() {
        EmployeeIDPage.EID.removeEmployee(employee);
    }
}
```

Figure 24

addEmployee and removeEmployee Classes Implementation

The following Figure 25 shows the implementation of `ParticipantArrayList` and `EmployeeArrayList` classes. The classes execute the methods that are responsible for modifying arraylists by adding and removing participants and employees.

```

public class ParticipantArrayList {

    ArrayList as = new ArrayList();
    public void addParticipant(Participant p){
        as.add(p);
    }
    public void removeParticipant(Participant p){
        as.remove(p);
    }
}

public class EmployeeArrayList {

    ArrayList employee = new ArrayList();
    public void addEmployee(Employee e){
        employee.add(e);
    }

    public void removeEmployee(Employee e){
        employee.remove(e);
    }
}

```

Figure 25

ParticipantArrayList and EmployeeArrayList Implementations

As it is mentioned before, ManagerPage_participant class is used to run the part that deals with modifying participants. For example, the manager creates a request for adding participant. In ManagerPage_participant class, we create a Participant object. Then, the object is attached to AddParticipant command method. The Participant object handles the participant's data that will be added to the arraylist. After that, we create the invoker object in order to call execute_Manager_Commands() method, then, attach AddParticipant command object to the method. AddParticipant concrete command will be in charge of the requested command and send its result to ParticipantArrayList class, which is responsible for performing the action. Figure 26 shows that implementation of ManagerPage_participant class.

```

public class ManagerPage_participant {
    static ParticipantArrayList as = new ParticipantArrayList();

    private void addParticipantActionPerformed() {

        Manager_Commands addCus;
        Participant c;
        c = new Participant(/* passing participant data*/);

        addCus = new AddParticipant(c);
        invoker.execute_Manager_Commands(addCus);
    }

    private void deleteParticipantActionPerformed() {
        Manager_Commands deleteCus;

        for(int i = 0; i < as.getSize(); i++)
        {
            if (txt1.getText().equalsIgnoreCase(Integer.toString
                (as.getParticipant(i).getID())))
            {
                deleteCus = new RemoveParticipant(as.getParticipant(i));
                invoker.execute_Manager_Commands(deleteCus);
            }
            else
                continue;
        }
    }
}

```

Figure 26

ManagerPage_participant Class Implementation

Similarly, ManagerPage_employee class is used to run the part that deals with modifying employees. For example, the manager creates a request for removing an employee. Next, we create an Employee object. Then, the object is attached to removeEmployee command method. The Employee object handles the employee's data that will be removed from the employees' arraylist. After that, we create the invoker object and in order to call execute_Manager_Commands() method, then, attach removeEmployee command object to the method. The removeEmployee concrete command will be in charge of the requested command and send its result to EmployeeArrayList class, which is responsible for performing the action. Figure 27 shows that implementation of ManagerPage_employee class.

```

public class ManagerPage_employee {

    private void addEmployeeActionPerformed() {
        Manager_Commands addEmpl;

        e = new Employee(/*passing employee data*/);
        addEmpl = new addEmployee(e);
        invoker.execute_Manager_Commands(addEmpl);
    }

    private void deleteEmployeeActionPerformed() {
        Manager_Commands deleteEmpl;
        for(int i = 0; i < EmployeeIDPage.EID.Size(); i++)
        {
            if (EmployeeNum.getText().equalsIgnoreCase
                (EmployeeIDPage.EID.getEmployee(i).getID()))
            {
                deleteEmpl = new removeEmployee(EmployeeIDPage.EID.getEmployee(i));
                invoker.execute_Manager_Commands(deleteEmpl);
            }
            else
                continue;
        }
    }
}

```

Figure 27

ManagerPage_employee Class Implementation

3.2.4 Observer Design Pattern

The Observer Design Pattern comes under the Behavioral Patterns. It is used when any object called Observer wants to be automatically notified and updated when another object called Observable changes its state. Thus, Observer pattern is used to assure dependency between objects [12]. For example, when data exists in a database and can be displayed in different formats, such as a graph or a table, the Observer pattern allows the display classes to register themselves with the class that handle the data. Therefore, they can update their displays when they receive notifications about data changes [10]. Next, Figure 28 represents the Observer Pattern Structure.

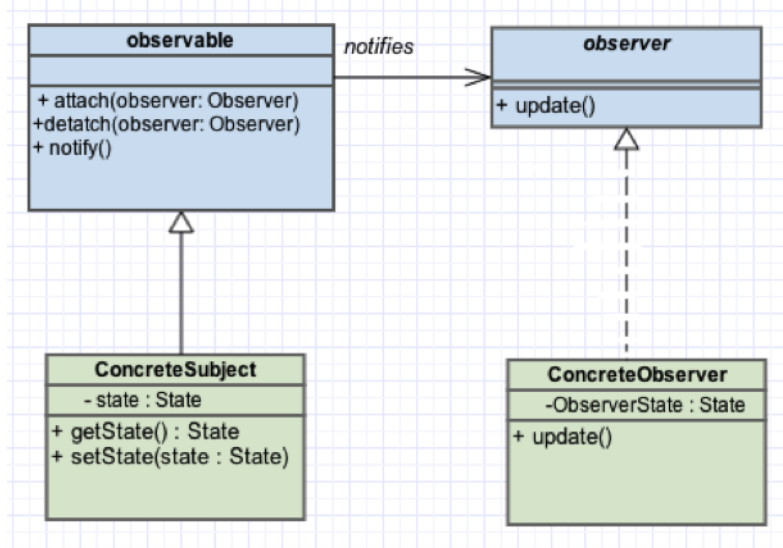


Figure 28

Observer Pattern Structure [4], [12]

The Observer Pattern Structure represented in Figure 28 includes different classes, which are:

- **Observable:** an interface or abstract class for adding and removing observers [12].
- **ConcreteSubject or ConcreteObservable:** it implements methods that are used by observers to determine the nature of the change. It also preserves the state of the object, and if an event happens, ConcreteSubject notifies the registered observers with the changes [12].
- **Observer:** an interface that defines the operations, which will be used to notify observers if any change occurs [12].
- **ConcreteObserver:** implements the Observer interface and make a respond to changes using the information represent the nature of the change received from the ConcreteSubject class [12].

3.2.4.1 Implementing Observer Pattern in the Application

The Observer Pattern is used to notify the employees with any updates that occur in the workshops details. The updates include the changes in the workshops fees, periods, and the number of available places. Figure 29 shows the Observer Pattern Structure that is used in the application.

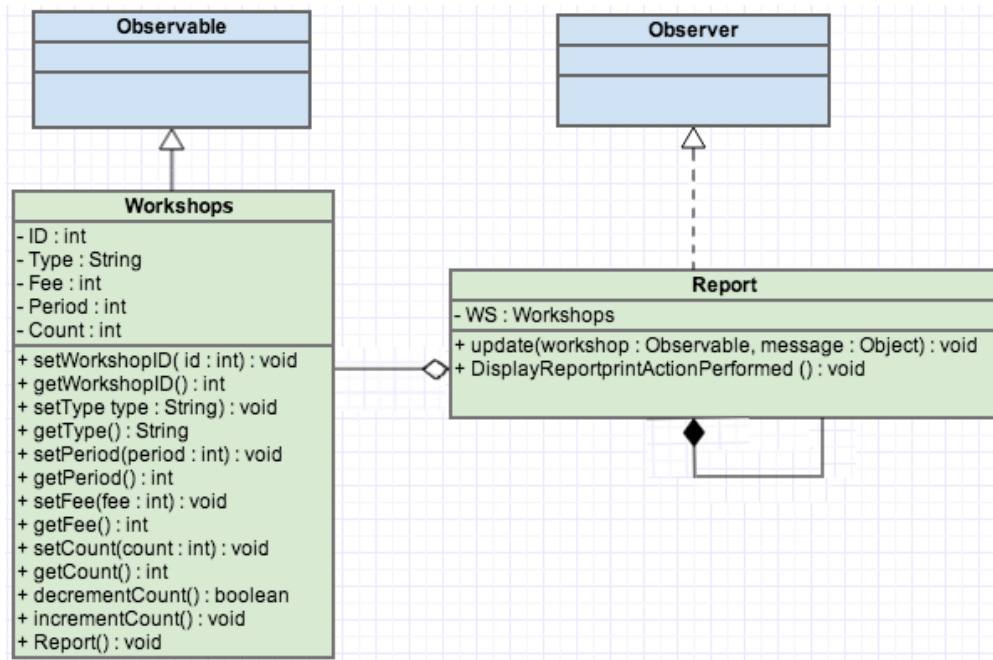


Figure 29

Observer Pattern Structure Used in the Application

As in Figure 29, Workshops class represents ConcreteSubject that implements Observable. Report class, which observes the changes in the state of the Workshops object, is created by implementing Observer interface. Observable and Observer are implemented using library classes, which provide the same methods needed for implementing Observable and Observer classes. The library for Observable class is `java.util.Observable` [13]. Furthermore, the library for Observer class is `java.util.Observer` [14].

In Workshops class represented in Figure 30, if there is a change that occurs in a workshop's places available, period, or fee, the observable object invokes its `setChanged()` method to show that there is a change in the workshop. After that, the observable object invokes its `notifyObservers()` method in order to update the registered observers with the changes.

```
public class Workshops extends Observable{
    int Fee, Period, Count;

    public int getPeriod()
    {
        return Period;
    }

    public int getFee()
    {
        return Fee;
    }

    // to count the number of places in workshops
    public int getCount()
    {
        return Count;
    }

    public void Report()
    {
        if (this.getWorkshopID()==1)
        {
            setChanged();
            notifyObservers("Computer Graphics");
        }
    }
}
```

Figure 30

Workshops Class Implementation Part 2

In Report class, `update()` method is called when the observable object changes its state and declares the changes by calling its `notifyObservers()` method. The observer should then use the observable object to determine and display its changes. Report class is also used to run Observer Pattern.

The method `DisplayReportprintActionPerformed()` that is defined in `Report` class demonstrates how to add an instance of the `Report` class to the observers' list. This can be done by attaching the `Report` instance to `addObserver()` method. For `Report` class implementation detail, see Figure 31.

```
public class Report implements Observer{

    static Workshops ComputerGraphics = new Workshops(1, "Computer Graphics",
    200, 10, 21);

    public void update(Observable workshop, Object message)
    {
        Workshops WS = (Workshops)workshop;
        String msg = (String)message;

        if(msg.equalsIgnoreCase("Computer Graphics")){

            WS = ComputerGraphics;

            String txt = "The number of places available: ";
            String txt2 = "" + WS.Count;
            String txt3 = "The fee of the workshop: ";
            String txt4 = "" + WS.getFee();
            String txt5 = "The period of the workshop: ";
            String txt6 = "" + WS.getPeriod();

            // display the report in a new message box
            MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5, txt6);
        }
    }

    private void DisplayReportprintActionPerformed() {
        Report report =new Report();

        if(WorkshopMenue.getSelectedIndex() == 0)
        {
            ComputerGraphics.addObserver(report);
            ComputerGraphics.Report();
        }
    }
}
```

Figure 31

Reports Class Implementation

The `Reports` class implementation represented in Figure 31 shows only one `Workshops` object example, which is the `Computer Graphics` workshop. The choice of the workshop report is retrieved using a `Combo Box`. See Figure 32 for the `Combo Box` used for different workshop reports.

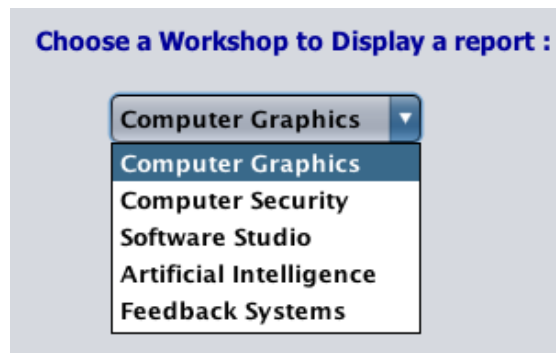


Figure 32

A Combo Box Providing a List of Workshop Reports

Next, Figure 33 represents the overall structure of design patterns usage in Workshop Management System application.

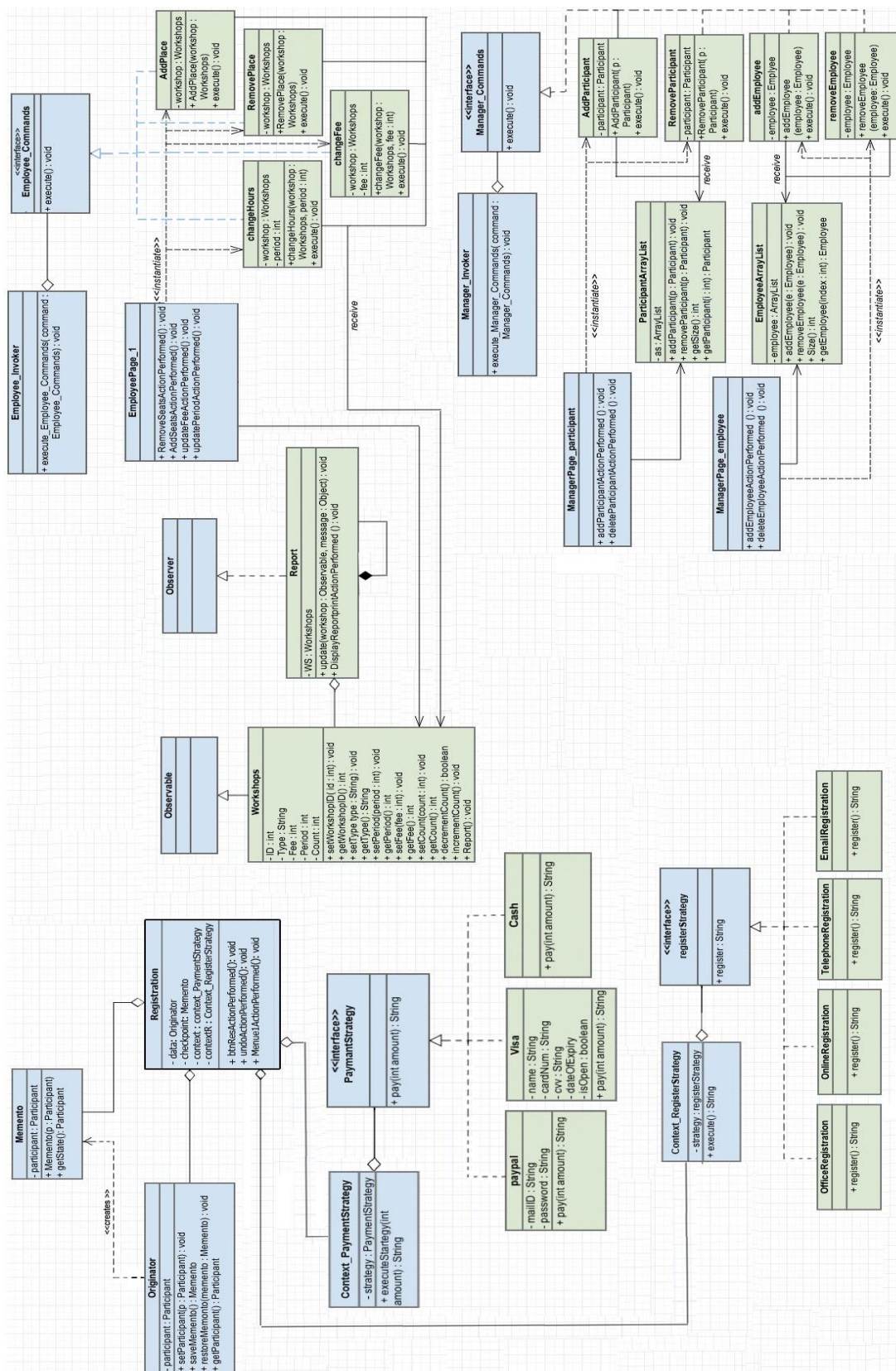


Figure 33

The Overall Workshop Management System Application Structure

Chapter 4: TESTING WORKSHOP MANAGEMENT SYSTEM APPLICATION

Testing Workshop Management System application allows the users to understand how to use the application. The following steps show how this program works:

1. After running the program, the user clicks Start button.
2. Three choices will display. The user can choose to display Participant, Employee or Manager page as shown in Figure 34.



Figure 34

Application Home Page

3. If the user chooses Participant page, a new window
4. displays as in Figure 35. The new window gives the user three options, such as New Registration, Cancel Registration, and Home.
 - New Registration: allows the user to make a new workshop registration.

- Cancel Registration: allows the user to cancel an existing workshop registration.
- Home: allows the user to get back to home page.



Figure 35

Participant Page

5. If New Registration is selected, the user will be directed to a new window where he can make a new workshop registration as shown in Figure 36.

As in Figure 36, the user is allowed to choose from different workshops to be enroll in. If the workshop is full, the user will be asked to choose a different workshop. Also, the user is allowed to choose the way of registration from a menu as the square shows in Figure 36. This part implements the Strategy Pattern for different types of registrations. Each strategy returns a string that represents the registration instructions.

The screenshot shows a web application window titled "Participant Registration". It contains the following elements:

- Workshop Selection:** A dropdown menu labeled "Choose the workshop you want to enroll in :" with "Computer Graphics" selected.
- Workshop Details:**
 - The Fee of the Workshop : 200
 - The length of the Workshop per hour : 21
- Registration Method Selection:** A dropdown menu labeled "Choose the way of registration :". The menu is open, showing options: "Telephone registration" (highlighted), "Email registration", "Telephone registration", "Office registration", and "Online registration". A red rectangle highlights this dropdown menu.
- Workshop Description:** A scrollable text area titled "Workshop Description :" containing details about the "Computer Graphics" course, including topics (ray tracing, graphics pipeline, etc.), instructor (Quinones), day/time (Monday, 9:00 - 11:30 am), session dates (August 28, 2015 to January 24, 2015), and location (South Berkeley Senior Center, 2939 Ellis St, Minneapolis, MN 55646).
- Contact Information:** A block of text stating: "Dear participant, please contact Registration Services (320) 567-5674 (M-F 8am - 4:30pm, Sat. & Sun. 8am - 2pm) or CS Workshop office 320-432-1234."
- Navigation:** "Undo" and "Back" buttons at the bottom right.

Figure 36

New Registration Page

6. If the Online registration strategy is selected, the user needs to enter his information as shown in Figure 37. The user needs to enter the information correctly; otherwise, error messages will occur.

Participant Registration

Choose the workshop you want to enroll in : **Artificial Intelligence**

The Fee of the Workshop : **1000**

The length of the Workshop per hour : **37**

Choose the way of registration : **Online registration**

Please enter the following information:

First Name :

Last Name :

Street Address :

Country :

City : State:

Zip code :

Mobile# : * Enter 10 digits

Email Address :

Choose Payment Options :

☐ Cash ☒ Credit Card ☐ Paypal

Card number :

Name on Card :

CVV :

Expiration Date :

Workshop Description :

Artificial Intelligence

This course includes interactive demonstrations which are intended to stimulate interest and to help students gain intuition about how artificial intelligence methods work under a variety of circumstances.

Instructor
Patrick Winston

Day/Time
Tuesday, 8:00 - 9:30 am

Session Dates
April 28, 2015 to July 24, 2015

Location
South Berkeley Senior Center
7689 Ellis St
Minneapolis, MN 55463
(510) 981-5170

Undo

Register

Back

Figure 37

Using Online Registration Strategy and Credit Card Payment Strategy

As in Figure 37, the square displays different payment options. This part implements the Strategy pattern for different payment strategies. If Cash is selected, then the user can register directly. On the other hand, if either Credit Card or PayPal payment strategies is selected, the user will need to enter the payment information before making the registration. See Figure 37 and 38 for Credit Card and PayPal payments details.

Participant Registration

Choose the workshop you want to enroll in : **Artificial Intelligence**

The Fee of the Workshop : **1000**

The length of the Workshop per hour : **37**

Choose the way of registration : **Online registration**

Please enter the following information:

First Name :

Last Name :

Street Address:

Country :

City : State:

Zip code :

Mobile# : * Enter 10 digits

Email Address :

Choose Payment Options :

☐ Cash ☐ Credit Card ☒ Paypal

Email Address :

Password :

Workshop Description :

Artificial Intelligence

This course includes interactive demonstrations which are intended to stimulate interest and to help students gain intuition about how artificial intelligence methods work under a variety of circumstances.

Instructor
Patrick Winston

Day/Time
Tuesday, 8:00 – 9:30 am

Session Dates
April 28, 2015 to July 24, 2015

Location
South Berkeley Senior Center
7689 Ellis St
Minneapolis, MN 55463
(510) 981-5170

Figure 38

Using Online Registration Strategy and PayPal Payment Strategy

6. Every strategy returns a string with the payment details as shown in Figure 39.

\$1000 fee will be paid using Paypal

Your confirmation number is: 349

Figure 39

Registration Confirmation Detail Page

7. If the Confirm button is clicked, the user will be directed to the final window as shown in Figure 40. Then, the user will be added to the participants' list. On the other hand, if the user selects Back button, the registration will not be confirmed and the user will get back to the registration window. In the registration window, the user can use Undo button shown in Figure 38 to restore the information he already entered. Restoring the information of the user implements Memento Design Pattern.

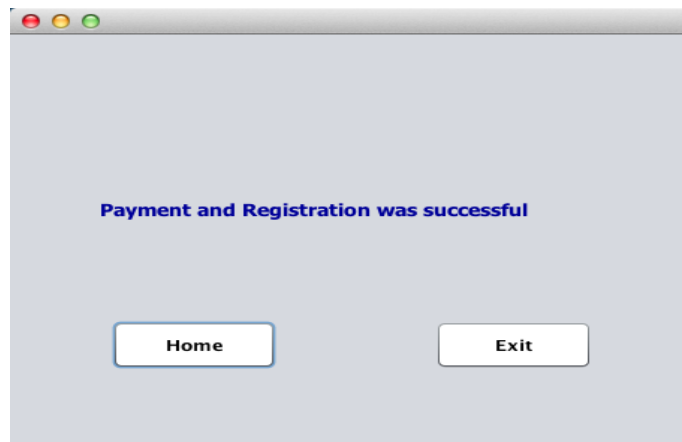


Figure 40

Final Window for Confirming Registration

8. Getting back to home page, if the Manager button is clicked, the manager will be directed to the Manager ID page shown in Figure 41. In this page, the manager needs to enter the correct ID number to be able to manage participants and employees information. Else, if the ID number is incorrect, an error message will occur.



Manager ID page

Manager Page
CS Workshop Management System

Enter your ID#

1111

Enter

Manage Participant information

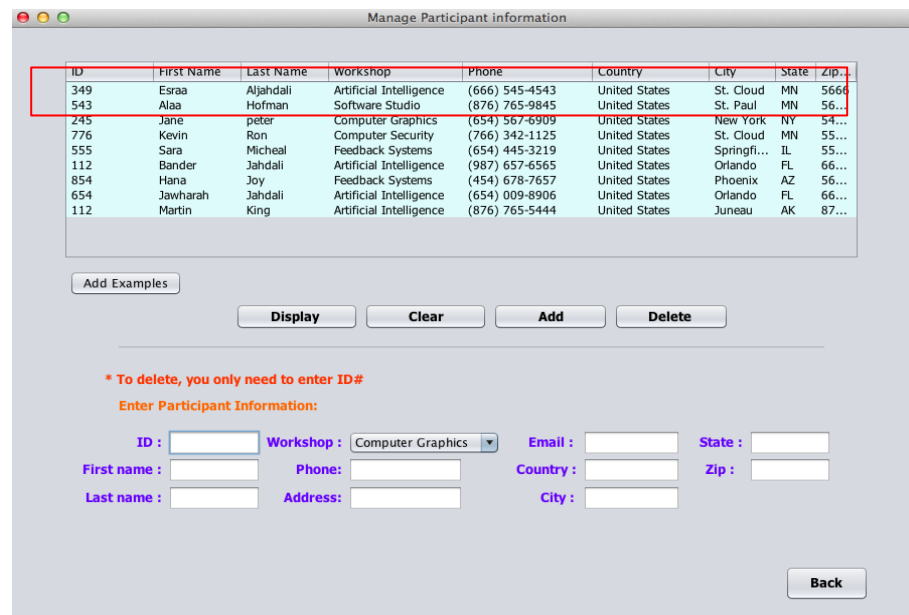
Manage Employee information

Back

Figure 41

Manager ID Page

9. If the manager chooses to manage participants' information, he/she will be directed to a new window where participants' information can be added, deleted, and displayed. See Figure 42 for details.



Manage Participant Information

ID	First Name	Last Name	Workshop	Phone	Country	City	State	Zip...
349	Esraa	Aljahdali	Artificial Intelligence	(666) 545-4543	United States	St. Cloud	MN	5666
543	Alaa	Hofman	Software Studio	(876) 765-9845	United States	St. Paul	MN	56...
245	Jane	peter	Computer Graphics	(654) 567-6909	United States	New York	NY	54...
776	Kevin	Ron	Computer Security	(766) 342-1125	United States	St. Cloud	MN	55...
555	Sara	Micheal	Feedback Systems	(654) 445-3219	United States	Springfi...	IL	55...
112	Bander	Jahdali	Artificial Intelligence	(987) 657-6565	United States	Orlando	FL	66...
854	Hana	Joy	Feedback Systems	(454) 678-7657	United States	Phoenix	AZ	56...
654	Jawharah	Jahdali	Artificial Intelligence	(654) 009-8906	United States	Orlando	FL	66...
112	Martin	King	Artificial Intelligence	(876) 765-5444	United States	Juneau	AK	87...

Add Examples

Display Clear Add Delete

* To delete, you only need to enter ID#

Enter Participant Information:

ID : Workshop : Email : State :

First name : Phone: Country : Zip :

Last name : Address: City :

Back

Figure 42

Manage Participant Information Page

10. The red square in Figure 42 shows how a participant is added to the participants' list in the manager page. The Display button is used to display the list of participant and Clear button is used to clear the same list. Figure 38 and 39 represent the participant registration details entered by the user. To add a new participant, the manager should enter the participant information correctly; otherwise, error messages will occur. Moreover, to delete a participant, the manager will need to enter the participant ID number and click Delete button. The manager must enter correct ID numbers that does not appear in the participants' list in case of addition, or the ID numbers should exist in the participants' list in case of deletion, otherwise, error messages will occur. A successful deletion process is shown in Figure 43. Adding and removing participants are implemented using Command design pattern.

Country	City	State	Zip...
United States	St. Cloud	MN	5666
United States	St. Paul	MN	56...
United States	New York	NY	54...
United States	St. Cloud	MN	55...
United States	Springfi...	IL	55...
United States	Orlando	FL	66...
United States	Phoenix	AZ	56...
United States	Orlando	FL	66...
United States	Juneau	AK	87...

Enter Participant Information:

ID : 349 Workshop : Computer Graphics Email : State :
 First name : Phone: Country : Zip :
 Last name : Address: City :

Figure 43

Successful Participant Deletion by the Manager

11. To cancel registration, the participant can click on Cancel Registration button shown in Figure 35. Then, the participant will be directed to a new window where he/she can enter his ID number and display the registration information as shown in Figure 44. After that, the participant can cancel the registration. The participant must enter the correct ID number; otherwise, an error message will occur.



Cancel Registration

Enter your registration ID# :

555

Display ID Information

First name: Sara
Last Name: Micheal
Workshop enrollment: Feedback Systems
Phone: (654) 445-3219
Address:
Email: sara@yahoo.com
Country: United States
City: Springfield
State: IL
Zip Code: 55123

Cancel Registration

Back

Figure 44

Participant Canceling Registration

12. If the participant cancels registration, he/she will be deleted from the participants' list represented in the manager page as shown in Figure 45.

Manage Participant information

ID	First Name	Last Name	Workshop	Phone	Country	City	State	Zip code
543	Alaa	Hofman	Software St...	(876) 765-...	United States	St. Paul	MN	56444
245	Jane	peter	Computer G...	(654) 567-...	United States	New York	NY	54443
776	Kevin	Ron	Computer S...	(766) 342-...	United States	St. Cloud	MN	55434
112	Bander	Jahdali	Artificial Int...	(987) 657-...	United States	Orlando	FL	66543
854	Hana	Joy	Feedback S...	(454) 678-...	United States	Phoenix	AZ	56675
654	Jawharah	Jahdali	Artificial Int...	(654) 009-...	United States	Orlando	FL	66543
112	Martin	King	Artificial Int...	(876) 765-...	United States	Juneau	AK	87870

Add Examples

Display Clear Add Delete

*** To delete, you only need to enter ID#**

Enter Participant Information:

ID : Workshop : Email : State :

First name : Phone: Country : Zip :

Last name : Address: City :

Back

Figure 45

Participant Deleted from Participants' List in Manager Page

13. Getting back to the manager ID page shown in Figure 41, the manager can manage employees' information. By clicking on Manage Employee Information button, the manager will be directed to a new window where the employees' information can be added, deleted, and displayed as shown in Figure 46. If the manager wants to add a new employee, he/she should enter the employee's information correctly; otherwise, error messages will occur. On the other hand, if the manager wants to delete an employee, he/she will need to enter the employee ID number and click the Delete button. Adding and removing employees are implemented using Command Design Pattern. See Figure 46 for an example of a successful addition of an employee to the system.

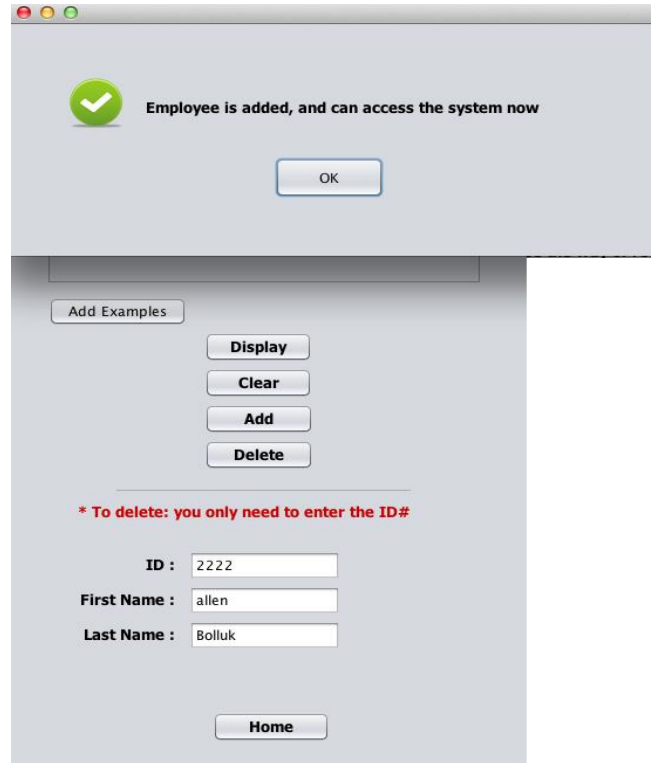


Figure 46

Successful Employee Addition Process by the Manager

14. Add Example button in Figure 46 is used to add some employee examples in the list of employee in order to modify them later. If the employee is added to the system, he/she will be listed in the employees' list represented in the manager page as shown in Figure 47. The employee then will be able to access the system and modify workshops' information. Else, if the employee is deleted, he/she will not be able to access the system and make modification.

ID	First Name	Last Name
1234	Robin	Nelson
5678	Mike	Stone
9876	Ron	Doom
6655	Susan	Mike
2211	Nancy	Moore
2222	allen	Bolluk

Add Examples

Display

Clear

Add

Delete

*** To delete: you only need to enter the ID#**

ID : 2222

First Name : allen

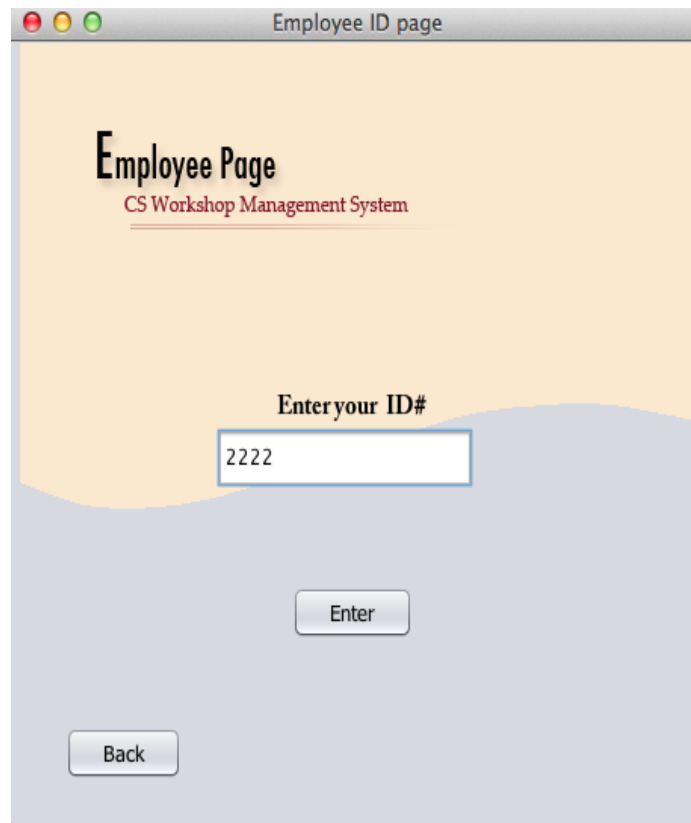
Last Name : Bolluk

Home

Figure 47

Employee Addition to Employees' List in Manager Page

15. Getting back to home page, if Employee button is selected, the employee will be directed to the Employee ID page shown in Figure 48. In this page, the employee needs to enter the correct ID number to be able to manage workshops' information. Else, if the ID number is incorrect, an error message will occur.



The screenshot shows a web application window titled "Employee ID page". The page has a light orange header area with the text "Employee Page" in a large, bold, black font, and "CS Workshop Management System" in a smaller, red font below it. The main content area has a light blue background. It contains a label "Enter your ID#" above a text input field. The input field contains the number "2222". Below the input field is a button labeled "Enter". At the bottom left of the page is a button labeled "Back".

Figure 48

Employee ID Page

16. If the employee enters the correct ID number, he/she will be directed to a new window where the available workshops' information can be modified, such as add places, remove places, update workshop fee, and update workshop period. These operations are implemented using Command Design Pattern. See Figure 49 for details.

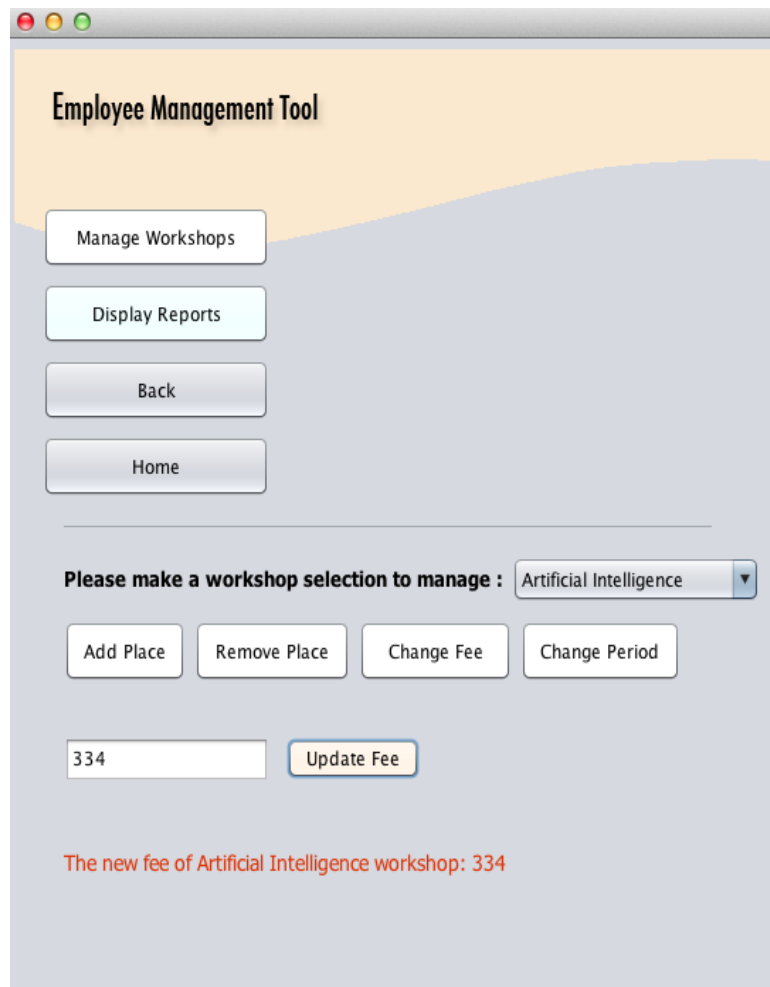


Figure 49

Employee Page for Workshops Modification

17. Besides making modification to the workshops' information, the employee can display reports of the workshops by clicking on Display Reports button shown in Figure 49. The employee will then be directed to a new window where different workshops reports can be chosen as represented in Figure 50.



Figure 50

Employee Page for Displaying Workshop Reports

18. After choosing a workshop, the employee clicks on Display Report button to be directed to the reports window. In the reports window, the employee can observe the changes that occurred in the workshop's information as shown in Figure 51. Displaying the changes that occur in the workshops' information is implemented using Observer design pattern.

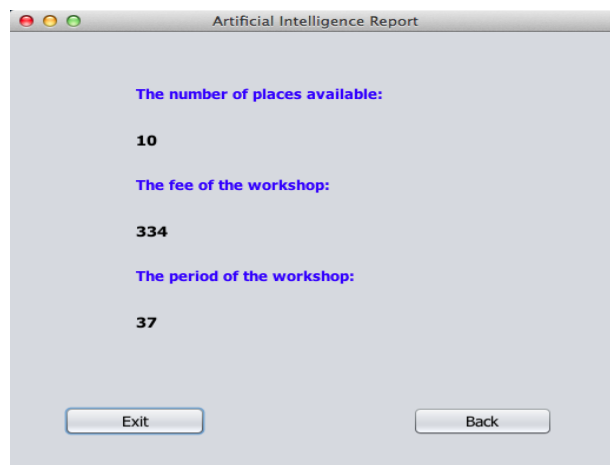


Figure 51

Reports Window Displaying Workshops Changes

4.1 Application Improvement Possibilities

There are many ways to improve an application. However, since this paper is concentrating on the Workshop Management System application and design patterns, it is better to focus on improving the use of design patterns in the application.

1. One way to improve the application is by extending the Strategy Pattern concrete classes. We can increase the number of payment and registration strategies, which give the users the flexibility to choose the payment option and registration method that meet their preference.
2. Moreover, it is possible to extend the command pattern concrete classes. For example, providing the manager with additional commands. This allows the manager to have greater control over the system.
3. Another way to make improvements to the application is by adding more observer concrete classes. This allows more users to get notified by the changes in the system. In the application, the employee is the one that uses Report class to be updated with the system changes. We can also add the manager and participants as observers; thus, they can check the changes in the available workshops' information.

Chapter 5: IMPORTANCE OF DESIGN PATTERNS

There are several benefits of using design patterns while solving design problems. And as we implemented an application that applies and combines several patterns, the advantages of design patterns now are more sensible. The following represent some of these advantages:

1. Writing code is flexible since design patterns provide a clear picture of how to implement the design of the solution [4], [6].
2. Handling the code modification by supporting the idea of code reusability. The changes are accommodated using several techniques, such as composition, delegation, and other techniques that design patterns provide [4], [6].
3. Design patterns represent proven techniques that are widely used. Thus, this helps to reduce the technical risks while building the system solution [4], [6].
4. Design patterns can be applied to any language since they are language neutral. Accordingly, this helps the developers to apply design patterns using the coding language they prefer and have more knowledge about [15].
5. Since there are many patterns that solve different types of problems, design patterns are highly flexible and can be applied in any type of application or domain [4], [6].
6. Patterns with different structures can cooperate with each other within the same system to solve different problems as represented in the Workshop Management System application.

5.1 Conclusion

Design patterns are one of the most powerful methods for establishing large software systems. They help developers find solutions to reoccurring problems that they encounter [3]. And for patterns to be effective and deliver the best design experience, it is important that design patterns can be used by less experienced designers [16]. This helps most of the software designers to achieve successful design experience [16]. From this point, developing Workshop Management System application was less challenging because of the use of design patterns. The structures and methods represented by design patterns provided a significant amount of flexibility and extensibility in creating the application. Because of the program idea, choosing Strategy, Memento, Observer, and Command design patterns to be used in the application was reasonable. These patterns were suitable to the application requirements. The application is implemented for demonstration purpose of combining design patterns. However, it can be improved later to be used in commercial fields since design patterns used in the application accommodate the changes that might be applied to the system.

References

- [1] P. Kushana, *Software Architecture Design Pattern in Java*. Boston, MA: Auerbach Publications, 2004.
- [2] M. Grand, *Java Enterprise Design Pattern*. New York, NY: Wiley, 2002.
- [3] M. P. Selvan and K. S. Swarup, "Development of power flow software using design patterns," *IEEE Transactions on Power Systems*, vol. 21, May 2006, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1626365&refinements%3D4291944246%26queryText%3Ddesign-patterns>. [Accessed March 6, 2015].
- [4] E. Freeman, E. Robson, K. Sierra, and B. Bates, *Head Firsts Design Patterns*. Sebastopol, CA: O'Reilly, 2004.
- [5] J. Dong, K. Zhang, and S. Yang, "Visualizing design patterns in their applications and compositions," *IEEE Transactions on Software Engineering*, vol. 33, p. 433, July 2007, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4227827> [Accessed March 20, 2015].
- [6] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA: Addison Wesley, 1995.
- [7] S. Dasculu, N. Hao, and N. Debnath, "Design patterns automation with template library," in *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*, December 2005, p. 700, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1577183>. [Accessed March 11, 2015].
- [8] J. M. Bieman, D. Jain, and Helen J. Yang, "OO design patterns, design structure, and program changes: An industrial case study," in *Proceedings of the IEEE International Conference on Software Maintenance*, 2001, p. 580, 2001, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=972775>. [Accessed February 11, 2015].
- [9] C. Chung, *Pro Objective-C Design Patterns for iOS*. New York, NY: Apress, 2011.
- [10] S. J. Metsker, *Design Patterns Java Workbook*. Boston, MA: Addison Wesley, 2008.
- [11] J. Bishop, *C# 3.0 Design Patterns*. Sebastopol, CA: O'Reilly, 2008.
- [12] M. Huaxin and J. Shuai, "Design patterns in software development," in *IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, July

- 2011, pp. 322-323, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5982228>. [Accessed March 9, 2015].
- [13] “Class observable,” <http://docs.oracle.com/javase/7/docs/api/java/util/Observable.html>. [Accessed March 10, 2015].
- [14] “Interface observer,” <http://docs.oracle.com/javase/7/docs/api/java/util/Observer.html>. [Accessed March 10, 2015].
- [15] D. Manoblescu, W. Kozacynski, A. Miller, and J. Hogg, “The growing divide in the patterns world,” *IEEE Software*, vol. 24, p. 64, July 2007, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4267604>. [Accessed March 25, 2015].
- [16] C. Zhang and D. Budgen, “What do we know about the effectiveness of software design patterns?,” *IEEE Transactions on Software Engineering*, vol. 38, p. 1, August 2012, <http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=5975176>. [Accessed January 3, 2015].
- [17] Gliffy, Inc., “Diagram like a boss,” <http://www.gliffy.com/>. [Accessed March 23, 2015].
- [18] “NetBeans learning Java-resources,” <https://netbeans.org/kb/articles/learn-java.html>. [Accessed February 23, 2015].
- [19] “Java SE downloads,” <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. [Accessed February 23, 2015].

Appendix

Registration.java

```
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.util.Random;

public class Registration extends javax.swing.JFrame {

    public Registration() {
        initComponents();
        setTitle("Participant Registration");

        jLabel6.setVisible(false);
        Menuel.setVisible(false);
        countryList.setSelectedIndex(-1);
        Menuel.setSelectedIndex(-1);
        Menue.setSelectedIndex(-1);
        btnRes.setVisible(false);
        countryList.setVisible(false);
        txtFname.setVisible(false);
        txtLname1.setVisible(false);
        txtPrice.getText();
        jLabel8.setVisible(false);
        mobileB.setVisible(false);
        streetAddress.setVisible(false);
        emailB.setVisible(false);
        cityB.setVisible(false);
        stateB.setVisible(false);
        zipB.setVisible(false);
        jLabel3.setVisible(false);
        jLabel9.setVisible(false);
        jLabel10.setVisible(false);
        jLabel14.setVisible(false);
        jLabel39.setVisible(false);
        jLabel13.setVisible(false);
        jLabel4.setVisible(false);
        jLabel40.setVisible(false);
        jLabel12.setVisible(false);
        jLabel7.setVisible(false);
        jLabel12.setVisible(false);
        jLabel2.setVisible(false);
        cash.setVisible(false);
        visa.setVisible(false);
    }
}
```

```

        jLabel23.setVisible(false);
        txtvisa2.setVisible(false);
        jLabel22.setVisible(false);
        txtvisa3.setVisible(false);
        jLabel20.setVisible(false);
        txtvisa1.setVisible(false);
        jLabel24.setVisible(false);
        txtpaypal.setVisible(false);
        jLabel21.setVisible(false);
        jPasswordField2.setVisible(false);
        Paypal.setVisible(false);
        jSeparator1.setVisible(false);
        jSeparator3.setVisible(false);
        jSeparator4.setVisible(false);
        jLabel5.setVisible(false);
        txtvisa.setVisible(false);
    }

    //-----
    public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }
    //-----
    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel1 = new javax.swing.JLayeredPane();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        txtFname = new javax.swing.JTextField();
        jLabel9 = new javax.swing.JLabel();
        txtLname1 = new javax.swing.JTextField();
        stateB = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();

```



```

streetAddress = new javax.swing.JTextField();
jLabel12 = new javax.swing.JLabel();
emailB = new javax.swing.JTextField();
cityB = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
zipB = new javax.swing.JTextField();
jLayeredPane3 = new javax.swing.JLayeredPane();
jLabel11 = new javax.swing.JLabel();
Menue2 = new javax.swing.JComboBox();
jLabel29 = new javax.swing.JLabel();
jLabel30 = new javax.swing.JLabel();
jLabel31 = new javax.swing.JLabel();
txtFname2 = new javax.swing.JTextField();
jLabel32 = new javax.swing.JLabel();
jLabel33 = new javax.swing.JLabel();
txtPrice2 = new javax.swing.JLabel();
txtNumDay3 = new javax.swing.JTextField();
jLabel34 = new javax.swing.JLabel();
caculate2 = new javax.swing.JButton();
txtLname3 = new javax.swing.JTextField();
jTextField14 = new javax.swing.JTextField();
jLabel35 = new javax.swing.JLabel();
jTextField15 = new javax.swing.JTextField();
jTextField16 = new javax.swing.JTextField();
jLabel36 = new javax.swing.JLabel();
jTextField17 = new javax.swing.JTextField();
jTextField18 = new javax.swing.JTextField();
jLabel37 = new javax.swing.JLabel();
jLabel38 = new javax.swing.JLabel();
jTextField19 = new javax.swing.JTextField();
jLabel39 = new javax.swing.JLabel();
jLabel40 = new javax.swing.JLabel();
mobileB = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
cash = new javax.swing.JRadioButton();
visa = new javax.swing.JRadioButton();
jLabel8 = new javax.swing.JLabel();
txtvisa = new javax.swing.JTextField();
message = new javax.swing.JLabel();
countyList = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
Menuel = new javax.swing.JComboBox();
Paypal = new javax.swing.JRadioButton();
jSeparator1 = new javax.swing.JSeparator();
jSeparator2 = new javax.swing.JSeparator();
jSeparator3 = new javax.swing.JSeparator();
jSeparator4 = new javax.swing.JSeparator();
jLabel20 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();

```

```

jLabel22 = new javax.swing.JLabel();
txtvisa1 = new javax.swing.JTextField();
txtvisa2 = new javax.swing.JTextField();
txtvisa3 = new javax.swing.JTextField();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
txtpaypal = new javax.swing.JTextField();
jPasswordField2 = new javax.swing.JPasswordField();
jLabel15 = new javax.swing.JLabel();
countryList = new javax.swing.JComboBox();
btnback = new javax.swing.JButton();
btnRes = new javax.swing.JButton();
description = new java.awt.TextArea();
jLabel15 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
Menue = new javax.swing.JComboBox();
txtPrice = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
Undo = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jLayeredPanel1.setBackground(new java.awt.Color(204, 255, 255));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel2.setText("Please enter the following information:");
jLayeredPanel1.add(jLabel2);
jLabel2.setBounds(0, 50, 290, 30);

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel3.setText("First Name :");
jLayeredPanel1.add(jLabel3);
jLabel3.setBounds(30, 90, 100, 30);

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel4.setText("State:");
jLayeredPanel1.add(jLabel4);
jLabel4.setBounds(290, 210, 100, 30);

txtFname.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtFnameActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtFname);
txtFname.setBounds(140, 90, 140, 30);

```

```

jLabel9.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel9.setText("Last Name :");
jLayeredPanel1.add(jLabel9);
jLabel9.setBounds(30, 120, 110, 30);

txtLname1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLname1ActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtLname1);
txtLname1.setBounds(140, 120, 140, 30);

stateB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        stateBActionPerformed(evt);
    }
});
jLayeredPanel1.add(stateB);
stateB.setBounds(340, 210, 60, 30);

jLabel10.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel10.setText("Street Address:");
jLayeredPanel1.add(jLabel10);
jLabel10.setBounds(30, 150, 100, 30);

streetAddress.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        streetAddressActionPerformed(evt);
    }
});
jLayeredPanel1.add(streetAddress);
streetAddress.setBounds(140, 150, 260, 30);

jLabel12.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel12.setText("Email Address :");
jLayeredPanel1.add(jLabel12);
jLabel12.setBounds(30, 300, 90, 30);

emailB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        emailBActionPerformed(evt);
    }
});
jLayeredPanel1.add(emailB);
emailB.setBounds(140, 300, 260, 30);

cityB.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cityBActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(cityB);
    cityB.setBounds(140, 210, 140, 30);

    jLabel13.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    jLabel13.setText("Zip code :");
    jLayeredPanel1.add(jLabel13);
    jLabel13.setBounds(30, 240, 90, 30);

    jLabel14.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    jLabel14.setText("Country :");
    jLayeredPanel1.add(jLabel14);
    jLabel14.setBounds(30, 180, 90, 40);

    zipB.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            zipBActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(zipB);
    zipB.setBounds(140, 240, 140, 30);

    jLabel29.setText("Enter the following information:");
    jLayeredPane3.add(jLabel29);
    jLabel29.setBounds(20, 50, 210, 30);

    jLabel30.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    jLabel30.setText("First Name :");
    jLayeredPane3.add(jLabel30);
    jLabel30.setBounds(30, 90, 100, 30);

    jLabel31.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    jLabel31.setText("State:");
    jLayeredPane3.add(jLabel31);
    jLabel31.setBounds(290, 210, 100, 30);

    txtFname2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtFname2ActionPerformed(evt);
        }
    });

    txtPrice2.setBorder(javax.swing.BorderFactory.createLineBorder(new
    java.awt.Color(0, 51, 102)));
    jLayeredPane3.add(txtPrice2);
    txtPrice2.setBounds(290, 400, 140, 30);

```

```

txtNumDay3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNumDay3ActionPerformed(evt);
    }
});
jLayeredPane3.add(txtNumDay3);
txtNumDay3.setBounds(300, 360, 140, 30);

jLabel34.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel34.setText("Last Name :");
jLayeredPane3.add(jLabel34);
jLabel34.setBounds(30, 120, 110, 30);

jLayeredPane3.add(caculate2);
caculate2.setBounds(210, 360, 75, 29);

txtLname3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLname3ActionPerformed(evt);
    }
});
jLayeredPane3.add(txtLname3);
txtLname3.setBounds(140, 120, 140, 30);

jTextField14.setText("jTextField1");
jTextField14.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField14ActionPerformed(evt);
    }
});
jLayeredPane3.add(jTextField14);
jTextField14.setBounds(340, 210, 60, 30);

jLabel35.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel35.setText("Street Address:");
jLayeredPane3.add(jLabel35);
jLabel35.setBounds(30, 150, 100, 30);

jTextField15.setText("jTextField1");
jTextField15.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField15ActionPerformed(evt);
    }
});
jLayeredPane3.add(jTextField15);
jTextField15.setBounds(140, 150, 360, 30);

```

```

jTextField16.setText("jTextField1");
jTextField16.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField16ActionPerformed(evt);
    }
});
jLayeredPane3.add(jTextField16);
jTextField16.setBounds(140, 180, 140, 30);

jLabel36.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel36.setText("Mobile# :");
jLayeredPane3.add(jLabel36);
jLabel36.setBounds(30, 260, 90, 40);

jTextField17.setText("jTextField1");
jTextField17.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField17ActionPerformed(evt);
    }
});
jLayeredPane3.add(jTextField17);
jTextField17.setBounds(140, 270, 140, 30);

jTextField18.setText("jTextField1");
jTextField18.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField18ActionPerformed(evt);
    }
});
jLayeredPane3.add(jTextField18);
jTextField18.setBounds(140, 210, 140, 30);

jLabel37.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel37.setText("Zip code :");
jLayeredPane3.add(jLabel37);
jLabel37.setBounds(30, 240, 90, 30);

jLabel38.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel38.setText("City :");
jLayeredPane3.add(jLabel38);
jLabel38.setBounds(30, 210, 90, 40);

jTextField19.setText("jTextField1");
jTextField19.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField19ActionPerformed(evt);
    }
});

```

```

    }
});
jLayeredPane3.add(jTextField19);
jTextField19.setBounds(140, 240, 140, 30);

jLayeredPanel1.add(jLayeredPane3);
jLayeredPane3.setBounds(0, 0, 0, 0);

jLabel39.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel39.setText("City :");
jLayeredPanel1.add(jLabel39);
jLabel39.setBounds(30, 210, 90, 40);

jLabel40.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel40.setText("Mobile# :");
jLayeredPanel1.add(jLabel40);
jLabel40.setBounds(30, 270, 90, 30);

mobileB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mobileBActionPerformed(evt);
    }
});
jLayeredPanel1.add(mobileB);
mobileB.setBounds(140, 270, 140, 30);

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel7.setText("Choose Payment Options :");
jLayeredPanel1.add(jLabel7);
jLabel7.setBounds(20, 360, 180, 20);

cash.setText("Cash");
cash.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cashActionPerformed(evt);
    }
});
jLayeredPanel1.add(cash);
cash.setBounds(30, 390, 93, 20);

visa.setText("Credit Card");
visa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        visaActionPerformed(evt);
    }
});
jLayeredPanel1.add(visa);
visa.setBounds(110, 390, 103, 23);

jLabel8.setText(" Card number :");

```

```

jLayeredPanel1.add(jLabel8);
jLabel8.setBounds(30, 440, 100, 37);

txtvisa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtvisaActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtvisa);
txtvisa.setBounds(140, 440, 170, 30);

message.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
message.setForeground(new java.awt.Color(255, 0, 51));
jLayeredPanel1.add(message);
message.setBounds(270, 620, 310, 20);

countyList.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
countyList.setForeground(new java.awt.Color(51, 51, 255));
jLayeredPanel1.add(countyList);
countyList.setBounds(50, 60, 400, 260);

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel6.setText("Choose the way of registration :");
jLayeredPanel1.add(jLabel6);
jLabel6.setBounds(0, 0, 223, 30);

Menuel.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
Menuel.setForeground(new java.awt.Color(0, 102, 204));
Menuel.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
    "Email registration", "Telephone registration", "Office registration",
    "Online registration", " " }));
Menuel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        MenuelActionPerformed(evt);
    }
});
jLayeredPanel1.add(Menuel);
Menuel.setBounds(230, 0, 205, 30);

Paypal.setText("Paypal");
Paypal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        PaypalActionPerformed(evt);
    }
});
jLayeredPanel1.add(Paypal);
Paypal.setBounds(220, 390, 72, 23);
jLayeredPanel1.add(jSeparator1);
jSeparator1.setBounds(230, 510, 0, 12);
jLayeredPanel1.add(jSeparator2);

```



```

jSeparator2.setBounds(0, 40, 520, 12);
jLayeredPanel1.add(jSeparator3);
jSeparator3.setBounds(10, 420, 520, 12);
jLayeredPanel1.add(jSeparator4);
jSeparator4.setBounds(10, 340, 520, 12);

jLabel20.setText("Expiration Date :");
jLayeredPanel1.add(jLabel20);
jLabel20.setBounds(20, 530, 110, 30);

jLabel21.setText("Password :");
jLayeredPanel1.add(jLabel21);
jLabel21.setBounds(60, 470, 142, 37);

jLabel22.setText("CVV :");
jLayeredPanel1.add(jLabel22);
jLabel22.setBounds(90, 500, 40, 37);

txtvisa1.setText("MMYY");
txtvisa1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtvisa1ActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtvisa1);
txtvisa1.setBounds(140, 530, 60, 30);

txtvisa2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtvisa2ActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtvisa2);
txtvisa2.setBounds(140, 470, 170, 30);

txtvisa3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtvisa3ActionPerformed(evt);
    }
});
jLayeredPanel1.add(txtvisa3);
txtvisa3.setBounds(140, 500, 60, 30);

jLabel23.setText("Name on Card :");
jLayeredPanel1.add(jLabel23);
jLabel23.setBounds(30, 480, 142, 20);

jLabel24.setText("Email Address :");
jLayeredPanel1.add(jLabel24);
jLabel24.setBounds(30, 440, 110, 37);

```

```

    txtpaypal.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtpaypalActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(txtpaypal);
    txtpaypal.setBounds(140, 440, 260, 30);

    jPasswordField2.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jPasswordField2ActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(jPasswordField2);
    jPasswordField2.setBounds(140, 470, 170, 28);

    jLabel5.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel5.setForeground(new java.awt.Color(255, 51, 0));
    jLabel5.setText("* Enter 10 digits");
    jLayeredPanel1.add(jLabel5);
    jLabel5.setBounds(290, 280, 270, 14);

    countryList.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "United States", "Albania", "Algeria", "Andorra", "Angola",
"Antigua and Barbuda", "Argentina", "Afghanistan", "Armenia", "Aruba",
"Australia", "Austria", "Azerbaijan", "Bahamas, The", "Bahrain",
"Bangladesh", "Barbados", "Belarus", "Belgium", "Belize", "Benin",
"Bhutan", "Bolivia", "Bosnia and Herzegovina", "Botswana", "Brazil",
"Brunei ", "Bulgaria", "Burkina Faso", "Burma", "Burundi", "Cambodia",
"Cameroon", "Canada", "Cape Verde", "Central African Republic", "Chad",
"Chile", "China", "Colombia", "Comoros", "Congo, Democratic Republic of
the", "Congo, Republic of the", "Costa Rica", "Cote d'Ivoire", "Croatia",
"Cuba", "Curacao", "Cyprus", "Czech Republic", "Denmark", "Djibouti",
"Dominica", "Dominican Republic", "East Timor (see Timor-Leste)",
"Ecuador", "Egypt", "El Salvador", "Equatorial Guinea", "Eritrea",
"Estonia", "Ethiopia", "Finland", "France", "Gabon", "Gambia, The",
"Georgia", "Germany", "Ghana", "Greece", "Grenada", "Guatemala", "Guinea",
"Guinea-Bissau", "Guyana", "Haiti", "Holy See", "Honduras", "Hong Kong",
"Hungary", "celand", "India", "Indonesia", "Iran", "Iraq", "Ireland",
"Israel", "Italy", "Jamaica", "Japan", "Jordan", "Kazakhstan", "Kenya",
"Kiribati", "Korea, North", "Korea, South", "Kosovo", "Kuwait",
"Kyrgyzstan", "Laos", "Latvia", "Lebanon", "Lesotho", "Liberia", "Libya",
"Liechtenstein", "Lithuania", "Luxembourg", "adagascar", "Malawi",
"Malaysia", "Maldives", "Mali", "Malta", "Marshall Islands", "Mauritania",
"Mauritius", "Mexico", "Micronesia", "Moldova", "Monaco", "Mongolia",
"Montenegro", "Morocco", "Mozambique", "Nauru", "Nepal", "Netherlands",
"Netherlands Antilles", "New Zealand", "Nicaragua", "Niger", "Nigeria",
"North Korea", "Norway", "Oman", "Pakistan", "Palau", "Paraguay", "Peru",

```

```

"Philippines", "Poland", "Portugal", "Qatar", "San Marino", "Sao Tome and
Principe", "Saudi Arabia", "Senegal", "Serbia", "Seychelles", "Sierra
Leone", "Singapore", "Sint Maarten", "Slovakia", "Slovenia", "Solomon
Islands", "Somalia", "South Africa", "South Korea", "South Sudan", "Spain
", "Sri Lanka", "Sudan", "Suriname", "Swaziland ", "Sweden", "Switzerland",
"Syria", "Togo", "Tonga", "Trinidad and Tobago", "Tunisia", "Turkey",
"Turkmenistan", "Tuvalu", "Ukraine", "United Arab Emirates", "United
Kingdom", "Uruguay", "Uzbekistan", "Yemen" }));
countryList.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        countryListActionPerformed(evt);
    }
});
jLayeredPanel.add(countryList);
countryList.setBounds(140, 180, 260, 27);

btnback.setText("Back");
btnback.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnbackActionPerformed(evt);
    }
});

btnRes.setText("Register");
btnRes.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResActionPerformed(evt);
    }
});

jLabel15.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel15.setText("Workshop Description :");

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel1.setText("Choose the workshop you want to enroll in :");

Menue.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
Menue.setForeground(new java.awt.Color(0, 102, 204));
Menue.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Computer Graphics", "Artificial Intelligence", "Computer Security",
"Software Studio", "Feedback Systems" }));
Menue.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        MenueActionPerformed(evt);
    }
});

txtPrice.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
txtPrice.setForeground(new java.awt.Color(255, 0, 51));

```

```

jLabel17.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel17.setText("The Fee of the Workshop : ");

jLabel16.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel16.setText("The length of the Workshop per hour :");

jLabel19.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel19.setForeground(new java.awt.Color(255, 0, 51));

Undo.setText("Undo");
Undo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        UndoActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLayeredPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 559,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel15)
        .addGap(0, 0, Short.MAX_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(btnRes,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnback,
javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Undo,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(description,
javax.swing.GroupLayout.PREFERRED_SIZE, 382,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(40, 40, 40)))
        .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 290,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(Menuue,
javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()
        .addComponent(jLabel17,
javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtPrice,
javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createSequentialGroup()
        .addComponent(jLabel16)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()
        .addGap(20, 20, 20)
        .addComponent(jLabel18,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 65,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel19))))))
        .addGap(21, 21, 21)))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(80, 80, 80)
            .addComponent(jLabel15)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(description,
javax.swing.GroupLayout.PREFERRED_SIZE, 454,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(Undo, javax.swing.GroupLayout.PREFERRED_SIZE,
41, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(btnRes,
javax.swing.GroupLayout.PREFERRED_SIZE, 43,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(btnback,
javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(40, 40, 40))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASE
LINE)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(Menu,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jLabel17,
        javax.swing.GroupLayout.PREFERRED_SIZE, 30,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txtPrice,
        javax.swing.GroupLayout.PREFERRED_SIZE, 30,
        javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel16,
        javax.swing.GroupLayout.PREFERRED_SIZE, 30,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel18)
    .addComponent(jLabel19))
    .addGap(18, 18, 18)
    .addComponent(jLayeredPanel1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 624,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents
//-----
private void btnbackActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_btnbackActionPerformed
    close();
    new Home().setVisible(true);
} // GEN-LAST: event_btnbackActionPerformed
//-----

Random r = new Random(); // to generate a random number
int rr = r.nextInt(400); // random number between 1 and 400
String cc = "";
static int selectType = 0;
static int selectPayment = 0;
String FName;
String Lname;
String WorkshopType;
Context_PaymentStrategy context;

```

```

static ParticipantArrayList as = new ParticipantArrayList();
static Originator data = new Originator();
static Memento checkpoint;

String Fphone;
String Faddress;
String Femail;
String Fcity;
String Fstate;
String Fcountry;
String Fzip;

//-----
private void btnResActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btnResActionPerformed

    //=====
    int index;

    if (Menu.getSelectedIndex() == 0) {

        selectType = 0;
        WorkshopType = "Computer Graphics";

        index = countryList.getSelectedIndex();

        if (Menu.getSelectedIndex() == -1) {

            ErrorMessage EM = new ErrorMessage("Please select a
workshop.");
            EM.setVisible(true);
        } else if (txtFname.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your first
name.");
            EM.setVisible(true);

        } else if (txtLname1.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your last
name.");
            EM.setVisible(true);

        } else if (mobileB.getText().isEmpty() ||
(mobileB.getText().length() != 10) || !(checkNumber(mobileB.getText()))) {

            ErrorMessage EM = new ErrorMessage("Please enter a valid a
phone number with 10 digits");

```



```

        EM.setVisible(true);

    } else if (streetAddress.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your
address");
        EM.setVisible(true);

    } else if (emailB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter you Email
address");
        EM.setVisible(true);

    } else if (cityB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
city");
        EM.setVisible(true);

    } else if (stateB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
state");
        EM.setVisible(true);

    } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
country");
        EM.setVisible(true);

    } else if (zipB.getText().isEmpty() || (zipB.getText().length()
!= 5) || !(checkNumber(zipB.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter the zip
code with 5 digits");
        EM.setVisible(true);

    } else {

        Participant c = new Participant(rr, txtFname.getText(),
txtLname1.getText(), WorkshopType, mobileB.getText(),
streetAddress.getText(), emailB.getText(), cityB.getText(),
stateB.getText(), countryList.getSelectedItem().toString(), zipB.getText(),
index);

        if (cash.isSelected()) {
            context = new Context_PaymentStrategy(new Cash());
            String amountMsg =

```

```

(context.executeStrategy(Integer.parseInt(txtPrice.getText())));

    cc += rr;

    String sss = "  Your confirmation number is: " + rr;

    close();

    MessageBox mb = new MessageBox(sss, amountMsg, c);
    mb.setVisible(true);

    } else if (visa.isSelected()) {
        if (txtvisa2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Enter name on
card");

            EM.setVisible(true);
        } else if (txtvisa.getText().isEmpty() ||
!(checkNumber(txtvisa.getText())) {
            ErrorMessage EM = new ErrorMessage("Enter a valid
card number");

            EM.setVisible(true);
        } else if (txtvisa3.getText().isEmpty() ||
!(checkNumber(txtvisa3.getText())) {

            ErrorMessage EM = new ErrorMessage("Enter cvv
code");

            EM.setVisible(true);
        } else if (txtvisa1.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Enter the
expiration date");

            EM.setVisible(true);

        } else {

            context = new Context_PaymentStrategy(new
            Visa(txtvisa2.getText(), txtvisa.getText(), txtvisa3.getText(),
            txtvisa1.getText()));

            String amountMsg =
            (context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            cc += rr;
            String sss = "Your confirmation number is: " + rr;

            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);
        }
    } else if (Paypal.isSelected()) {
        if (txtpaypal.getText().isEmpty()) {

```

```

        ErrorMessage EM = new ErrorMessage("Please enter
your PayPal email");
        EM.setVisible(true);
    } else if (jPasswordField2.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
email password");
        EM.setVisible(true);
    } else {
        context = new Context_PaymentStrategy(new
Paypal(txtpaypal.getText(), jPasswordField2.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));

        cc += rr;
        String sss = "Your confirmation number is: " + rr;

        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }
} else {
    ErrorMessage EM = new ErrorMessage("Please choose the
way of payment.");
    EM.setVisible(true);
    // message1.setText("Please choose the way of
payment");
}

// creates a memento containing a snapshot of its current
internal state.
data.setParticipant(c);

//stores internal state of the Originator object
checkpoint = data.saveMemento();
}

} //End if Computer Graphics

//=====
if (Menu.getSelectedIndex() == 1) {
    selectType = 1;
    WorkshopType = "Artificial Intelligence";
    index = countryList.getSelectedIndex();

    if (Menu.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
workshop.");
    }
}

```

```

        EM.setVisible(true);
    } else if (txtFname.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your first
name.");
        EM.setVisible(true);

    } else if (txtLname1.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your last
name.");
        EM.setVisible(true);

    } else if (mobileB.getText().isEmpty() ||
(mobileB.getText().length() != 10) || !(checkNumber(mobileB.getText()))) {

        ErrorMessage EM = new ErrorMessage("Please enter a valid a
phone number with 10 digits");
        EM.setVisible(true);

    } else if (streetAddress.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your
address");
        EM.setVisible(true);

    } else if (emailB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter you Email
address");
        EM.setVisible(true);

    } else if (cityB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
city");
        EM.setVisible(true);

    } else if (stateB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
state");
        EM.setVisible(true);

    } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
country");
        EM.setVisible(true);

```

```

        } else if (zipB.getText().isEmpty() || (zipB.getText().length()
!= 5) || !(checkNumber(zipB.getText())))) {

            ErrorMessage EM = new ErrorMessage("Please enter the zip
code with 5 digits");
            EM.setVisible(true);

        } else {
            Participant c = new Participant(rr, txtFname.getText(),
txtLname1.getText(), WorkshopType, mobileB.getText(),
streetAddress.getText(), emailB.getText(), cityB.getText(),
stateB.getText(), countryList.getSelectedItem().toString(), zipB.getText(),
index);

            if (cash.isSelected()) {
                context = new Context_PaymentStrategy(new Cash());
                String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
                Fname = txtFname.getText();
                Lname = txtLname1.getText();
                // price = Integer.parseInt(txtPrice.getText());
                cc += rr;
                String sss = "Your confirmation number is: " + rr;

                close();
                MessageBox mb = new MessageBox(sss, amountMsg, c);
                mb.setVisible(true);
            } else if (visa.isSelected()) {
                if (txtvisa2.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Enter name on
card");
                    EM.setVisible(true);
                } else if (txtvisa.getText().isEmpty() ||
!(checkNumber(txtvisa.getText())))) {
                    ErrorMessage EM = new ErrorMessage("Enter a vaild
card number");
                    EM.setVisible(true);
                } else if (txtvisa3.getText().isEmpty() ||
!(checkNumber(txtvisa3.getText())))) {

                    ErrorMessage EM = new ErrorMessage("Enter cvv
code");
                    EM.setVisible(true);
                } else if (txtvisa1.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Enter the
expiration date");
                    EM.setVisible(true);
                } else {

```

```

        context = new Context_PaymentStrategy(new
Visa(txtvisa2.getText(), txtvisa.getText(), txtvisa3.getText(),
txtvisa1.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
        cc += rr;
        String sss = "Your confirmation number is: " + rr;
        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }
    } else if (Paypal.isSelected()) {
        if (txtpaypal.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
your PayPal email");
            EM.setVisible(true);
        } else if (jPasswordField2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
email password");
            EM.setVisible(true);
        } else {
            context = new Context_PaymentStrategy(new
Paypal(txtpaypal.getText(), jPasswordField2.getText()));
            String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            cc += rr;
            String sss = "Your confirmation number is: " + rr;

            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);
        }
    } else {
        ErrorMessage EM = new ErrorMessage("Please choose the
way of payment.");
        EM.setVisible(true);
    }

    data.setParticipant(c);
    checkpoint = data.saveMemento();
}
} //End if Artificial Intelligence

//=====
if (Menue.getSelectedIndex() == 2) {
    selectType = 2;
    WorkshopType = "Computer Security";
    index = countryList.getSelectedIndex();
}

```

```

        if (Menue.getSelectedIndex() == -1) {

            ErrorMessage EM = new ErrorMessage("Please select a
workshop.");
            EM.setVisible(true);
        } else if (txtFname.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your first
name.");
            EM.setVisible(true);

        } else if (txtLname1.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your last
name.");
            EM.setVisible(true);

        } else if (mobileB.getText().isEmpty() ||
(mobileB.getText().length() != 10) || !(checkNumber(mobileB.getText()))) {

            ErrorMessage EM = new ErrorMessage("Please enter a valid a
phone number with 10 digits");
            EM.setVisible(true);

        } else if (streetAddress.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your
address");
            EM.setVisible(true);

        } else if (emailB.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter you Email
address");
            EM.setVisible(true);

        } else if (cityB.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter the
city");
            EM.setVisible(true);

        } else if (stateB.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter the
state");
            EM.setVisible(true);

        } else if (countryList.getSelectedIndex() == -1) {

```

```

        ErrorMessage EM = new ErrorMessage("Please select a
country");
        EM.setVisible(true);

        } else if (zipB.getText().isEmpty() || (zipB.getText().length()
!= 5) || !(checkNumber(zipB.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter the zip
code with 5 digits");
        EM.setVisible(true);

        } else {

        Participant c = new Participant(rr, txtFname.getText(),
txtLname1.getText(), WorkshopType, mobileB.getText(),
streetAddress.getText(), emailB.getText(), cityB.getText(),
stateB.getText(), countryList.getSelectedItem().toString(), zipB.getText(),
index);

        if (cash.isSelected()) {
            context = new Context_PaymentStrategy(new Cash());
            String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            Fname = txtFname.getText();
            Lname = txtLname1.getText();

            cc += rr;

            String sss = "Your confirmation number is: " + rr;
            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);
        } else if (visa.isSelected()) {
            if (txtvisa2.getText().isEmpty()) {

                ErrorMessage EM = new ErrorMessage("Enter name on
card");
                EM.setVisible(true);
            } else if (txtvisa.getText().isEmpty() ||
!(checkNumber(txtvisa.getText())) {

                ErrorMessage EM = new ErrorMessage("Enter a valid
card number");
                EM.setVisible(true);
            } else if (txtvisa3.getText().isEmpty() ||
!(checkNumber(txtvisa3.getText())) {

                ErrorMessage EM = new ErrorMessage("Enter cvv
code");

```



```

        EM.setVisible(true);
    } else if (txtvisa1.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Enter the
expiration date");
        EM.setVisible(true);

    } else {
        context = new Context_PaymentStrategy(new
Visa(txtvisa2.getText(), txtvisa.getText(), txtvisa3.getText(),
txtvisa1.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
        cc += rr;
        String sss = "Your confirmation number is:" + rr;

        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }

    } else if (Paypal.isSelected()) {
        if (txtpaypal.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
your PayPal email");
            EM.setVisible(true);
        } else if (jPasswordField2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
email password");
            EM.setVisible(true);
        } else {
            context = new Context_PaymentStrategy(new
Paypal(txtpaypal.getText(), jPasswordField2.getText()));
            String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            cc += rr;
            String sss = "Your confirmation number is: " + rr;

            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);
        }
    } else {
        ErrorMessage EM = new ErrorMessage("Please choose the
way of payment.");
        EM.setVisible(true);
    }
}

```

```

        data.setParticipant(c);
        checkpoint = data.saveMemento();
    }

    }//End if Computer Security
    //=====

    if (Menue.getSelectedIndex() == 3) {
        selectType = 3;
        WorkshopType = "Software Studio";
        index = countryList.getSelectedIndex();

        if (Menue.getSelectedIndex() == -1) {

            ErrorMessage EM = new ErrorMessage("Please select a
workshop.");
            EM.setVisible(true);
        } else if (txtFname.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your first
name.");
            EM.setVisible(true);

        } else if (txtLname1.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your last
name.");
            EM.setVisible(true);
        } else if (mobileB.getText().isEmpty() ||
(mobileB.getText().length() != 10) || !(checkNumber(mobileB.getText()))) {

            ErrorMessage EM = new ErrorMessage("Please enter a valid a
phone number with 10 digits");
            EM.setVisible(true);

        } else if (streetAddress.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter your
address");
            EM.setVisible(true);

        } else if (emailB.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter you Email
address");
            EM.setVisible(true);

        } else if (cityB.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter the

```

```

city");
        EM.setVisible(true);

    } else if (stateB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
state");
        EM.setVisible(true);

    } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
country");
        EM.setVisible(true);

    } else if (zipB.getText().isEmpty() || (zipB.getText().length()
!= 5) || !(checkNumber(zipB.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter the zip
code with 5 digits");
        EM.setVisible(true);

    } else {
        Participant c = new Participant(rr, txtFname.getText(),
txtLname1.getText(), WorkshopType, mobileB.getText(),
streetAddress.getText(), emailB.getText(), cityB.getText(),
stateB.getText(), countryList.getSelectedItem().toString(), zipB.getText(),
index);

        if (cash.isSelected()) {
            context = new Context_PaymentStrategy(new Cash());
            String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            Fname = txtFname.getText();
            Lname = txtLname1.getText();
            cc += rr;
            String sss = "Your confirmation number is:" + rr;

            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);
        } else if (visa.isSelected()) {
            if (txtvisa2.getText().isEmpty()) {

                ErrorMessage EM = new ErrorMessage("Enter name on
card");
                EM.setVisible(true);
            } else if (txtvisa.getText().isEmpty() ||
!(checkNumber(txtvisa.getText())) {
                ErrorMessage EM = new ErrorMessage("Enter a valid

```

```

card number");
        EM.setVisible(true);
    } else if (txtvisa3.getText().isEmpty() ||
!(checkNumber(txtvisa3.getText())) {

        ErrorMessage EM = new ErrorMessage("Enter cvv
code");

        EM.setVisible(true);
    } else if (txtvisa1.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Enter the
expiration date");

        EM.setVisible(true);

    } else {
        context = new Context_PaymentStrategy(new
Visa(txtvisa2.getText(), txtvisa.getText(), txtvisa3.getText(),
txtvisa1.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
        cc += rr;
        String sss = "Your confirmation number is:" + rr;

        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }
} else if (Paypal.isSelected()) {
    if (txtpaypal.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
your PayPal email");

        EM.setVisible(true);
    } else if (jPasswordField2.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
email password");

        EM.setVisible(true);
    } else {
        context = new Context_PaymentStrategy(new
Paypal(txtpaypal.getText(), jPasswordField2.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
        cc += rr;
        String sss = "Your confirmation number is: " + rr;

        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }
} else {

```

```

        ErrorMessage EM = new ErrorMessage("Please choose the
way of payment.");
        EM.setVisible(true);
    }

    data.setParticipant(c);
    checkpoint = data.saveMemento();
}

} //End if Software Studio
//=====

if (Menu.getSelectedIndex() == 4) {
    selectType = 4;
    WorkshopType = "Feedback Systems";
    index = countryList.getSelectedIndex();
    if (Menu.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
workshop.");
        EM.setVisible(true);
    } else if (txtFname.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your first
name.");
        EM.setVisible(true);

    } else if (txtLname1.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your last
name.");
        EM.setVisible(true);

    } else if (mobileB.getText().isEmpty() ||
(mobileB.getText().length() != 10) || !(checkNumber(mobileB.getText()))) {

        ErrorMessage EM = new ErrorMessage("Please enter a valid a
phone number with 10 digits");
        EM.setVisible(true);

    } else if (streetAddress.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter your
address");
        EM.setVisible(true);

    } else if (emailB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter you Email
address");

```

```

        EM.setVisible(true);

    } else if (cityB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
city");
        EM.setVisible(true);

    } else if (stateB.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter the
state");
        EM.setVisible(true);

    } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select a
country");
        EM.setVisible(true);

    } else if (zipB.getText().isEmpty() || (zipB.getText().length()
!= 5) || !(checkNumber(zipB.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter the zip
code with 5 digits");
        EM.setVisible(true);

    } else {
        Participant c = new Participant(rr, txtFname.getText(),
txtLname1.getText(), WorkshopType, mobileB.getText(),
streetAddress.getText(), emailB.getText(), cityB.getText(),
stateB.getText(), countryList.getSelectedItem().toString(), zipB.getText(),
index);

        if (cash.isSelected()) {
            context = new Context_PaymentStrategy(new Cash());
            String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
            Fname = txtFname.getText();
            Lname = txtLname1.getText();
            // price = Integer.parseInt(txtPrice.getText());
            cc += rr;
            String sss = "Your confirmation number is:" + rr;

            close();
            MessageBox mb = new MessageBox(sss, amountMsg, c);
            mb.setVisible(true);

        } else if (visa.isSelected()) {
            if (txtvisa2.getText().isEmpty()) {

```

```

        ErrorMessage EM = new ErrorMessage("Enter name on
card");
        EM.setVisible(true);
    } else if (txtvisa.getText().isEmpty() ||
!(checkNumber(txtvisa.getText())) {
        ErrorMessage EM = new ErrorMessage("Enter a valid
card number");
        EM.setVisible(true);
    } else if (txtvisa3.getText().isEmpty() ||
!(checkNumber(txtvisa3.getText())) {

        ErrorMessage EM = new ErrorMessage("Enter cvv
code");
        EM.setVisible(true);
    } else if (txtvisal.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Enter the
expiration date");
        EM.setVisible(true);

    } else {
        context = new Context_PaymentStrategy(new
Visa(txtvisa2.getText(), txtvisa.getText(), txtvisa3.getText(),
txtvisal.getText()));
        String amountMsg =
(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
        cc += rr;
        String sss = "Your confirmation number is:" + rr;

        close();
        MessageBox mb = new MessageBox(sss, amountMsg, c);
        mb.setVisible(true);
    }

} else if (Paypal.isSelected()) {
    if (txtpaypal.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
your PayPal email");
        EM.setVisible(true);
    } else if (jPasswordField2.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
email password");
        EM.setVisible(true);
    } else {
        context = new Context_PaymentStrategy(new
Paypal(txtpaypal.getText(), jPasswordField2.getText()));
        String amountMsg =

```

```

(context.executeStrategy(Integer.parseInt(txtPrice.getText())));
    cc += rr;
    String sss = "Your confirmation number is: " + rr;

    close();
    MessageBox mb = new MessageBox(sss, amountMsg, c);
    mb.setVisible(true);
}

} else {
    ErrorMessage EM = new ErrorMessage("Please choose the
way of payment.");
    EM.setVisible(true);
}

data.setParticipant(c);
checkpoint = data.saveMemento();
}
} //End if Feedback Systems

} //GEN-LAST:event_btnResActionPerformed
//-----
-----
private void visaActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_visaActionPerformed
    //=====

    // message1.setVisible(false);
    btnRes.setVisible(true);
    cash.setSelected(false);
    txtvisa.setVisible(true);
    jLabel8.setVisible(true);
    Paypal.setSelected(false);
    jLabel23.setVisible(true);
    txtvisa2.setVisible(true);
    jLabel22.setVisible(true);
    txtvisa3.setVisible(true);
    jLabel20.setVisible(true);
    txtvisa1.setVisible(true);
    jLabel24.setVisible(false);
    txtpaypal.setVisible(false);
    jLabel21.setVisible(false);
    jPasswordField2.setVisible(false);

    //=====
} //GEN-LAST:event_visaActionPerformed

static String select;

```



```

        private void txtFnameActionPerformed(java.awt.event.ActionEvent evt)
    {
        //GEN-FIRST:event_txtFnameActionPerformed
            // TODO add your handling code here:
        //GEN-LAST:event_txtFnameActionPerformed

        private void cashActionPerformed(java.awt.event.ActionEvent evt)
    {
        //GEN-FIRST:event_cashActionPerformed

            btnRes.setVisible(true);
            visa.setSelected(false);
            txtvisa.setVisible(false);
            jLabel8.setVisible(false);
            Paypal.setSelected(false);
            jLabel23.setVisible(false);
            txtvisa2.setVisible(false);
            jLabel22.setVisible(false);
            txtvisa3.setVisible(false);
            jLabel20.setVisible(false);
            txtvisa1.setVisible(false);
            jLabel24.setVisible(false);
            txtpaypal.setVisible(false);
            jLabel21.setVisible(false);
            jPasswordField2.setVisible(false);

            // TODO add your handling code here:
        //GEN-LAST:event_cashActionPerformed

        static boolean checkNumber(String number) {
            try {
                Integer.parseInt(number);
            } catch (NumberFormatException e) {
                return false;
            }
            return true;
        }

        private void MenueActionPerformed(java.awt.event.ActionEvent evt)
    {
        //GEN-FIRST:event_MenueActionPerformed

            if (Menue.getSelectedIndex() == 0) {

                if (Report.ComputerGraphics.Count == 0) {
                    ErrorMessage EM = new ErrorMessage("Computer Graphics
workshop is full, select another workshop");
                    EM.setVisible(true);
                } else {
                    jLabel6.setVisible(true);
                    Menue1.setVisible(true);
                }
            }
        }
    }

```

```

    }

    description.setText("Computer Graphics \n"
        + "\n"
        + "This course provides introduction to computer \n "
        + "graphics algorithms, software and hardware. \n"
        + "Topics include: ray tracing, the graphics
pipeline,\n"
        + " transformations, texture mapping, shadows,
sampling,\n"
        + " global illumination, splines, animation and
color.\n"
        + " This course offers 6 Engineering Design Points in
\n"
        + "MIT's EECS program.\n"
        + "\n"
        + "Instructor\n"
        + "Quinones \n"
        + "\n"
        + "Day/Time\n"
        + "Monday, 9:00 - 11:30 am\n"
        + "\n"
        + "Session Dates\n"
        + "August 28, 2015 to January 24, 2015\n"
        + "\n"
        + "\n"
        + "Location\n"
        + "South Berkeley Senior Center \n"
        + "2939 Ellis St\n"
        + "Minneapolis, MN 55646\n"
        + "(510) 981-5170 \n"
        + "\n");

txtPrice.setText(Integer.toString(Report.ComputerGraphics.getFee()));

// this is important for the messageBox
select = "Computer Graphics";

jLabel19.setText(Integer.toString(Report.ComputerGraphics.getPeriod()));

} else if (Menue.getSelectedIndex() == 1) {

    if (Report.ArtificialIntelligence.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Artificial Intelligence
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        jLabel6.setVisible(true);
        Menuel.setVisible(true);
    }
}

```

```

    }
    description.setText("Artificial Intelligence \n"
        + "\n"
        + "This course includes interactive demonstrations
which\n"
        + " are intended to stimulate interest and to help
students\n"
        + " gain intuition about how artificial intelligence
methods\n"
        + " work under a variety of circumstances. \n"
        + "\n"
        + "Instructor\n"
        + "Patrick Winston \n"
        + "\n"
        + "Day/Time\n"
        + "Tuesday, 8:00 - 9:30 am\n"
        + "\n"
        + "Session Dates\n"
        + "April 28, 2015 to July 24, 2015\n"
        + "\n"
        + "\n"
        + "Location\n"
        + "South Berkeley Senior Center \n"
        + "7689 Ellis St\n"
        + "Minneapolis, MN 55463\n"
        + "(510) 981-5170 \n"
        + "\n");

txtPrice.setText(Integer.toString(Report.ArtificialIntelligence.getFee()));
select = "Artificial Intelligence";

jLabel19.setText(Integer.toString(Report.ArtificialIntelligence.getPeriod()
));

} else if (Menue.getSelectedIndex() == 2) {

    if (Report.ComputerSecurity.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Computer Security
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        jLabel6.setVisible(true);
        Menuel.setVisible(true);
    }

    description.setText("Computer Security \n"
        + "\n"
        + "Investigation and detailed study of computer
security \n"

```

```

ensure data \n"
workshop will \n"
Windows\n"

+ " principles, mechanisms and implementations to
+ " protection and security of computers systems;
+ "focus on computer security issues in both UNIX and
+ " operating systems, database security. \n"
+ "\n"
+ "Instructor\n"
+ "Mark Newman \n"
+ "\n"
+ "Day/Time\n"
+ "Friday, 10:00 - 11:30 am\n"
+ "\n"
+ "Session Dates\n"
+ "August 28, 2015 to January 24, 2015\n"
+ "\n"
+ "\n"
+ "Location\n"
+ "South Berkeley Senior Center \n"
+ "3242 Ellis St\n"
+ "Minneapolis, MN 55463\n"
+ "(510) 981-5170 \n"
+ "\n");

txtPrice.setText(Integer.toString(Report.ComputerSecurity.getFee()));
select = "Computer Security";

jLabel19.setText(Integer.toString(Report.ComputerSecurity.getPeriod()));
} else if (Menue.getSelectedIndex() == 3) {

    if (Report.SoftwareStudio.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Software Studio
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        jLabel6.setVisible(true);
        Menuel.setVisible(true);
    }
    description.setText("Software Studio \n"
+ "\n"
+ "This course on software engineering covers design
\n"
+ " and implementation of medium-scale software
systems,\n"
+ " using web applications as a platform. In the
course, \n"
+ "students learn the fundamentals of structuring a web
\n"

```

```

+ "application and writing modular code, with an
emphasis on \n"
+ "conceptual design to achieve clarity, simplicity,
and modularity. \n"
+ " Topics also include functional programming,
relational \n"
+ "databases, and security.\n"
+ "\n"
+ "Instructor\n"
+ "Daniel Jackson  \n"
+ "\n"
+ "Day/Time\n"
+ "Monday, 12:00 - 1:30 am\n"
+ "\n"
+ "Session Dates\n"
+ "September 28, 2015 to October 24, 2015\n"
+ "\n"
+ "\n"
+ "Location\n"
+ "South Berkeley Senior Center  \n"
+ "8765 Ellis St\n"
+ "Minneapolis, MN 55463\n"
+ "(510) 981-5170 \n"
+ "\n");

txtPrice.setText(Integer.toString(Report.SoftwareStudio.getFee()));
select = "Software Studio";

jLabel19.setText(Integer.toString(Report.SoftwareStudio.getPeriod()));

} else if (Menue.getSelectedIndex() == 4) {

    if (Report.FeedbackSystems.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Feedback Systems
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        jLabel6.setVisible(true);
        Menuel.setVisible(true);
    }

    description.setText("Feedback Systems \n"
+ "\n"
+ "This course provides an introduction to the design
of \n "
+ "feedback systems. Topics covered include: properties
\n"
+ "and advantages of feedback systems, time-domain and
\n"

```

```

and \n"
criterion, \n"
application\n"
external \n"
compensation \n"
loops.\n"
+ " frequency-domain performance measures, stability
+ "degree of stability, root locus method, Nyquist
+ "frequency-domain design, compensation techniques,
+ " to a wide variety of physical systems, internal and
+ "compensation of operational amplifiers, modeling and
+ " of power converter systems, and phase lock
+ "\n"
+ "Instructor\n"
+ "Joel Dawson \n"
+ "\n"
+ "Day/Time\n"
+ "Friday, 12:00 - 1:30 am\n"
+ "\n"
+ "Session Dates\n"
+ "July 28, 2015 to September 24, 2015\n"
+ "\n"
+ "\n"
+ "Location\n"
+ "South Berkeley Senior Center \n"
+ "1234 Ellis St\n"
+ "Minneapolis, MN 55463\n"
+ "(510) 981-5170 \n"
+ "\n");

```

```

txtPrice.setText(Integer.toString(Report.FeedbackSystems.getFee()));
select = "Feedback Systems";

jLabel19.setText(Integer.toString(Report.FeedbackSystems.getPeriod()));
}
} //GEN-LAST:event_MenuActionPerformed

private void txtLnameActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txtLnameActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtLnameActionPerformed

private void txtvisaActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txtvisaActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtvisaActionPerformed

private void stateBActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_stateBActionPerformed

```

```

        // TODO add your handling code here:
    }//GEN-LAST:event_stateBActionPerformed

    private void streetAddressActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_streetAddressActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_streetAddressActionPerformed

    private void emailBActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_emailBActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_emailBActionPerformed

    private void cityBActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_cityBActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_cityBActionPerformed

    private void zipBActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_zipBActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_zipBActionPerformed

    private void Menue2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_Menue2ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_Menue2ActionPerformed

    private void txtFname2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txtFname2ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txtFname2ActionPerformed

    private void txtNumDay3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txtNumDay3ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txtNumDay3ActionPerformed

    private void caculate2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_caculate2ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_caculate2ActionPerformed

    private void txtLname3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txtLname3ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txtLname3ActionPerformed

    private void jTextField14ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jTextField14ActionPerformed

```

```

        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField14ActionPerformed

    private void jTextField15ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField15ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField15ActionPerformed

    private void jTextField16ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField16ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField16ActionPerformed

    private void jTextField17ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField17ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField17ActionPerformed
    private void jTextField18ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField18ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField18ActionPerformed

    private void jTextField19ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField19ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jTextField19ActionPerformed

    private void mobileBActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_mobileBActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_mobileBActionPerformed

    Context_RegisterStrategy contextR;

    private void MenuelActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_MenuelActionPerformed

        if (Menuel.getSelectedIndex() == 0) {

            selectPayment = 0;
            contextR = new Context_RegisterStrategy(new
            EmailRegistration());
            countyList.setText(contextR.execute());

            btnRes.setVisible(false);
            txtFname.setVisible(false);
            txtLname1.setVisible(false);
            mobileB.setVisible(false);
            streetAddress.setVisible(false);
            emailB.setVisible(false);

```



```

cityB.setVisible(false);
stateB.setVisible(false);
zipB.setVisible(false);
jLabel3.setVisible(false);
jLabel9.setVisible(false);
jLabel10.setVisible(false);
jLabel14.setVisible(false);
jLabel39.setVisible(false);
jLabel13.setVisible(false);
jLabel4.setVisible(false);
jLabel40.setVisible(false);
jLabel12.setVisible(false);
jLabel7.setVisible(false);
cash.setVisible(false);
visa.setVisible(false);
jLabel2.setVisible(false);
jSeparator3.setVisible(false);
jSeparator4.setVisible(false);
Paypal.setVisible(false);
jLabel5.setVisible(false);
countryList.setVisible(false);

} else if (Menuel.getSelectedIndex() == 1) {

    selectPayment = 1;
    contextR = new Context_RegisterStrategy(new
TelephoneRegistration());
    countyList.setText(contextR.execute());

    btnRes.setVisible(false);
    txtFname.setVisible(false);
    txtLname1.setVisible(false);
    mobileB.setVisible(false);
    streetAddress.setVisible(false);
    emailB.setVisible(false);
    cityB.setVisible(false);
    stateB.setVisible(false);
    zipB.setVisible(false);
    jLabel3.setVisible(false);
    jLabel9.setVisible(false);
    jLabel10.setVisible(false);
    jLabel14.setVisible(false);
    jLabel39.setVisible(false);
    jLabel13.setVisible(false);
    jLabel4.setVisible(false);
    jLabel40.setVisible(false);
    jLabel12.setVisible(false);
    jLabel7.setVisible(false);
    cash.setVisible(false);
    visa.setVisible(false);

```

```

        jLabel2.setVisible(false);
        jSeparator3.setVisible(false);
        jSeparator4.setVisible(false);
        Paypal.setVisible(false);
        jLabel5.setVisible(false);
        countryList.setVisible(false);

    } else if (Menuel.getSelectedIndex() == 2) {

        selectPayment = 2;
        contextR = new Context_RegisterStrategy(new
OfficeRegistration());
        countyList.setText(contextR.execute());

        btnRes.setVisible(false);
        txtFname.setVisible(false);
        txtLname1.setVisible(false);
        mobileB.setVisible(false);
        streetAddress.setVisible(false);
        emailB.setVisible(false);
        cityB.setVisible(false);
        stateB.setVisible(false);
        zipB.setVisible(false);
        jLabel3.setVisible(false);
        jLabel9.setVisible(false);
        jLabel10.setVisible(false);
        jLabel14.setVisible(false);
        jLabel39.setVisible(false);
        jLabel13.setVisible(false);
        jLabel4.setVisible(false);
        jLabel40.setVisible(false);
        jLabel12.setVisible(false);
        jLabel7.setVisible(false);
        cash.setVisible(false);
        visa.setVisible(false);
        jLabel2.setVisible(false);
        jSeparator3.setVisible(false);
        jSeparator4.setVisible(false);
        Paypal.setVisible(false);
        jLabel5.setVisible(false);
        countryList.setVisible(false);

    } else if (Menuel.getSelectedIndex() == 3) {

        selectPayment = 3;
        countyList.setVisible(false);
        contextR = new Context_RegisterStrategy(new
OnlineRegistration());
        // message2.setText(contextR.execute());

```

```

        txtFname.setVisible(true);
        txtLname1.setVisible(true);
        btnRes.setVisible(false);
        mobileB.setVisible(true);
        streetAddress.setVisible(true);
        emailB.setVisible(true);
        cityB.setVisible(true);
        stateB.setVisible(true);
        zipB.setVisible(true);
        jLabel3.setVisible(true);
        jLabel9.setVisible(true);
        jLabel10.setVisible(true);
        jLabel14.setVisible(true);
        jLabel39.setVisible(true);
        jLabel13.setVisible(true);
        jLabel4.setVisible(true);
        jLabel40.setVisible(true);
        jLabel12.setVisible(true);
        jLabel7.setVisible(true);
        cash.setVisible(true);
        visa.setVisible(true);
        jLabel2.setVisible(true);
        jSeparator3.setVisible(true);
        jSeparator4.setVisible(true);
        Paypal.setVisible(true);
        jLabel5.setVisible(true);
        countryList.setVisible(true);
    }
} //GEN-LAST:event_MenuelActionPerformed

private void PaypalActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_PaypalActionPerformed

    // message1.setVisible(false);
    btnRes.setVisible(true);
    visa.setSelected(false);
    cash.setSelected(false);
    txtvisa.setVisible(false);
    jLabel8.setVisible(false);
    jLabel23.setVisible(false);
    txtvisa2.setVisible(false);
    jLabel22.setVisible(false);
    txtvisa3.setVisible(false);
    jLabel20.setVisible(false);
    txtvisa1.setVisible(false);
    jLabel24.setVisible(true);
    txtpaypal.setVisible(true);
    jLabel21.setVisible(true);
    jPasswordField2.setVisible(true);

```

```

    }//GEN-LAST:event_PaypalActionPerformed

    private void txtvisa1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_txtvisa1ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_txtvisa1ActionPerformed

    private void txtvisa2ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_txtvisa2ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_txtvisa2ActionPerformed

    private void txtvisa3ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_txtvisa3ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_txtvisa3ActionPerformed

    private void txtpaypalActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_txtpaypalActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_txtpaypalActionPerformed

    private void jPasswordField2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jPasswordField2ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jPasswordField2ActionPerformed

    private void UndoActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_UndoActionPerformed

        btnRes.setVisible(true);
        Menue.setSelectedIndex(selectType);
        Menuel.setSelectedIndex(selectPayment);

        // uses the memento to restore its internal state.
        data.restoreMemento(checkpoint);

        txtFname.setText(data.getParticipant().getFame());
        txtLname1.setText(data.getParticipant().getLame());
        streetAddress.setText("" + data.getParticipant().getAddress());
        emailB.setText("" + data.getParticipant().getEmail());
        cityB.setText("" + data.getParticipant().getCity());
        stateB.setText("" + data.getParticipant().getState());

        countryList.setSelectedIndex(data.getParticipant().getCountryIndex());
        zipB.setText("" + data.getParticipant().getZip());
        mobileB.setText("" + data.getParticipant().getPhone());

```

```

    }//GEN-LAST:event_UndoActionPerformed

    private void countryListActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_countryListActionPerformed

    }//GEN-LAST:event_countryListActionPerformed
//-----
-----

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JComboBox Menue;
private javax.swing.JComboBox Menue1;
private javax.swing.JComboBox Menue2;
private javax.swing.JRadioButton Paypal;
private javax.swing.JButton Undo;
private javax.swing.JButton btnRes;
private javax.swing.JButton btnback;
private javax.swing.JButton caculate2;
private javax.swing.JRadioButton cash;
private javax.swing.JTextField cityB;
private javax.swing.JComboBox countryList;
private javax.swing.JLabel countyList;
private java.awt.TextArea description;
private javax.swing.JTextField emailB;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel36;

```

```

private javax.swing.JLabel jLabel37;
private javax.swing.JLabel jLabel38;
private javax.swing.JLabel jLabel39;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel40;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JLayeredPane jLayeredPanel1;
private javax.swing.JLayeredPane jLayeredPane3;
private javax.swing.JPasswordField jPasswordField2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JLabel message;
private javax.swing.JTextField mobileB;
private javax.swing.JTextField stateB;
private javax.swing.JTextField streetAddress;
private javax.swing.JTextField txtFname;
private javax.swing.JTextField txtFname2;
private javax.swing.JTextField txtLname1;
private javax.swing.JTextField txtLname3;
private javax.swing.JTextField txtNumDay3;
private javax.swing.JLabel txtPrice;
private javax.swing.JLabel txtPrice2;
private javax.swing.JTextField txtpaypal;
private javax.swing.JTextField txtvisa;
private javax.swing.JTextField txtvisa1;
private javax.swing.JTextField txtvisa2;
private javax.swing.JTextField txtvisa3;
private javax.swing.JRadioButton visa;
private javax.swing.JTextField zipB;
// End of variables declaration//GEN-END:variables
}

```

CancelRegistration.java

```

/*

```

```

    * To change this template, choose Tools | Templates
    * and open the template in the editor.
    */
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;

/**
 *
 * @author Esraa Aljahdali
 */
public class CancelRegistration extends javax.swing.JFrame {

    /**
     * Creates new form CancelRes
     */
    public CancelRegistration() {
        initComponents();
        setTitle("Cancel Registration");
    }

    public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel1 = new javax.swing.JLayeredPane();
        btncancel = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        btnResNum = new javax.swing.JTextField();
        btnback = new javax.swing.JButton();
        jLayeredPane2 = new javax.swing.JLayeredPane();
        displ = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();

```

```

jTextArea1 = new javax.swing.JTextArea();
txt = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

btncancel.setBackground(new java.awt.Color(242, 193, 193));
btncancel.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btncancel.setText("Cancel Registration");
btncancel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btncancelActionPerformed(evt);
    }
});
jLayeredPanel1.add(btncancel);
btncancel.setBounds(140, 440, 180, 30);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 171, 171));
jLabel1.setText("☞Enter your registration ID# :");
jLayeredPanel1.add(jLabel1);
jLabel1.setBounds(140, 40, 210, 20);

btnResNum.setBackground(new java.awt.Color(132, 206, 206));
btnResNum.setForeground(new java.awt.Color(255, 255, 255));
btnResNum.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResNumActionPerformed(evt);
    }
});
jLayeredPanel1.add(btnResNum);
btnResNum.setBounds(160, 70, 150, 30);

btnback.setBackground(new java.awt.Color(246, 241, 241));
btnback.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnback.setText("Back");
btnback.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnbackActionPerformed(evt);
    }
});
jLayeredPanel1.add(btnback);
btnback.setBounds(180, 470, 100, 30);
jLayeredPanel1.add(jLayeredPane2);
jLayeredPane2.setBounds(370, 100, 90, 110);

disply.setBackground(new java.awt.Color(205, 240, 240));
disply.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
disply.setText("Display ID Information");
disply.addActionListener(new java.awt.event.ActionListener() {

```



```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            displyActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(disply);
    disply.setBounds(140, 110, 190, 30);

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jScrollPane1.setViewportView(jTextArea1);

    jLayeredPanel1.add(jScrollPane1);
    jScrollPane1.setBounds(100, 160, 260, 240);

    txt.setFont(new java.awt.Font("Lucida Grande", 1, 13)); // NOI18N
    txt.setForeground(new java.awt.Color(243, 30, 73));
    jLayeredPanel1.add(txt);
    txt.setBounds(80, 410, 320, 20);

    javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLayeredPanel1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 470, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLayeredPanel1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 529, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold> //GEN-END: initComponents

static ParticipantArrayList as = new ParticipantArrayList();

private void btnbackActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btnbackActionPerformed
    dispose();
    new ParticipantPage().setVisible(true);

} //GEN-LAST:event_btnbackActionPerformed

private void btncancelActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btncancelActionPerformed

```

```

        for (int i = 0; i < ManagerPage_participant.as.getSize(); i++) {
            if ((Integer.parseInt(btnResNum.getText())) ==
ManagerPage_participant.as.getParticipant(i).getID()) {

ManagerPage_participant.as.removeParticipant(ManagerPage_participant.as.get
Participant(i));

                SuccessMessage SM = new SuccessMessage("Your registration
is canceled successfully");
                SM.setVisible(true);
            }
        }

    } //GEN-LAST:event_btncancelActionPerformed

    private void btnResNumActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btnResNumActionPerformed

        } //GEN-LAST:event_btnResNumActionPerformed

    private void displyActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_displyActionPerformed

        int count = 0;
        ErrorMessage em;

        // check if the input is integers
        try {
            Integer.parseInt(btnResNum.getText());
        } catch (NumberFormatException e) {
            em = new ErrorMessage(" Please integers only! \n");
            em.setVisible(true);
        }

        for (int i = 0; i < ManagerPage_participant.as.getSize(); i++) {
            int ID = Integer.parseInt(btnResNum.getText());

            if (ID == ManagerPage_participant.as.getParticipant(i).getID())
{

                count++;

                jTextArea1.setText("First name: " +
ManagerPage_participant.as.getParticipant(i).getFame()
                    + "\nLast Name: " +
ManagerPage_participant.as.getParticipant(i).getLame()
                    + "\nWorkshop enrollment: " +
ManagerPage_participant.as.getParticipant(i).getWorkshopType()
                    + "\nPhone: " +
ManagerPage_participant.as.getParticipant(i).getPhone()

```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
```

```

*/
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author Esraa Aljahdali
 */
public class ManagerIDPage extends javax.swing.JFrame {

    /**
     * Creates new form EmployeePage
     */
    public ManagerIDPage() {
        initComponents();
        addID();
        setTitle("Manager ID page");
        manageParticipant.setVisible(false);
        manageEmployee.setVisible(false);
    }

    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }
    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel1 = new javax.swing.JLayeredPane();
        jLabel1 = new javax.swing.JLabel();
        ManagerNum = new javax.swing.JTextField();
        ErrorMessage = new javax.swing.JLabel();

```

```

jButton1 = new javax.swing.JButton();
manageParticipant = new javax.swing.JButton();
manageEmployee = new javax.swing.JButton();
Back = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Adobe Caslon Pro", 1, 15)); //
NOI18N
jLabel1.setText("Enter your ID#");
jLayeredPanel.add(jLabel1);
jLabel1.setBounds(170, 200, 120, 20);

ManagerNum.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ManagerNumActionPerformed(evt);
    }
});
jLayeredPanel.add(ManagerNum);
ManagerNum.setBounds(140, 220, 170, 34);

ErrorMessage.setFont(new java.awt.Font("Adobe Caslon Pro", 1, 14));
// NOI18N
ErrorMessage.setForeground(new java.awt.Color(237, 54, 54));
jLayeredPanel.add(ErrorMessage);
ErrorMessage.setBounds(120, 260, 250, 30);

jButton1.setText("Enter");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jLayeredPanel.add(jButton1);
jButton1.setBounds(170, 300, 110, 30);

manageParticipant.setBackground(new java.awt.Color(255, 204, 204));
manageParticipant.setText("Manage Participant information");
manageParticipant.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        manageParticipantActionPerformed(evt);
    }
});
jLayeredPanel.add(manageParticipant);
manageParticipant.setBounds(110, 340, 230, 30);

manageEmployee.setBackground(new java.awt.Color(204, 204, 255));

```

```

        manageEmployee.setText("Manage Employee information");
        manageEmployee.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageEmployeeActionPerformed(evt);
            }
        });
        jLayeredPanel1.add(manageEmployee);
        manageEmployee.setBounds(110, 370, 230, 30);

        Back.setText("Back");
        Back.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BackActionPerformed(evt);
            }
        });
        jLayeredPanel1.add(Back);
        Back.setBounds(40, 430, 77, 32);

        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/managerPic.
gif"))); // NOI18N
        jLayeredPanel1.add(jLabel3);
        jLabel3.setBounds(0, -30, 460, 550);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 460,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 495,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            );

        pack();
    } // </editor-fold> // GEN-END: initComponents

```

```

ManagerArrayList MID = new ManagerArrayList();

public void addID()
{
    MID.addManager("1111");
    MID.addManager("2222");
    MID.addManager("3333");
}
private void addEmActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_addEmActionPerformed

    //GEN-LAST:event_addEmActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed

    for(int i = 0; i < MID.size(); i++)
    {
        if (ManagerNum.getText().equalsIgnoreCase(MID.getManager(i)))
        {
            manageParticipant.setVisible(true);
            manageEmployee.setVisible(true);
            ErrorMessage.setVisible(false);
        }
        else
            continue;
    }
    ErrorMessage.setText("ID does not exist, Please try again!");
    /*
    ErrorMessage.setText("ID does not exist, Please try again!"); */
//GEN-LAST:event_jButton1ActionPerformed

private void ManagerNumActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_ManagerNumActionPerformed

    //GEN-LAST:event_ManagerNumActionPerformed

private void
manageParticipantActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_manageParticipantActionPerformed
    close();
    new ManagerPage_participant().setVisible(true);
    //GEN-LAST:event_manageParticipantActionPerformed

private void manageEmployeeActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_manageEmployeeActionPerformed
    close();

```

```

        new ManagerPage_employee().setVisible(true);
    }//GEN-LAST:event_manageEmployeeActionPerformed

    private void BackActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_BackActionPerformed
        close();
        new Home().setVisible(true);
    } //GEN-LAST:event_BackActionPerformed

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton Back;
    private javax.swing.JLabel ErrorMessage;
    private javax.swing.JTextField ManagerNum;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLayeredPane jLayeredPanel;
    private javax.swing.JButton manageEmployee;
    private javax.swing.JButton manageParticipant;
    // End of variables declaration//GEN-END:variables
}

```

ManagerPage_employee.java

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Esraa_SP_Project;

import java.awt.Toolkit;
import javax.swing.table.DefaultTableModel;
import java.awt.event.WindowEvent;

/**
 *
 * @author esraaaljahdali
 */
public class ManagerPage_employee extends javax.swing.JFrame {

```



```

/**
 * Creates new form ManagerPage_employee
 */

    DefaultTableModel model;

    public ManagerPage_employee() {
        initComponents();
        setTitle("Manage Employee information");
        model = (DefaultTableModel)table.getModel();
    }

    public void addID()
    {
        EmployeeIDPage.EID.addEmployee(new Employee("1234", "Robin",
"NELSON"));
        EmployeeIDPage.EID.addEmployee(new Employee("5678", "Mike",
"STONE"));
        EmployeeIDPage.EID.addEmployee(new Employee("9876", "Ron",
"DOORN"));
        EmployeeIDPage.EID.addEmployee(new Employee("6655", "Susan",
"MIKE"));
        EmployeeIDPage.EID.addEmployee(new Employee("2211", "Nancy",
"MOORE"));
    }

    public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }

/**
 * This method is called from within the constructor to initialize the
form.
 * WARNING: Do NOT modify this code. The content of this method is
always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jScrollPane2 = new javax.swing.JScrollPane();
        table = new javax.swing.JTable();
        deleteEmployee = new javax.swing.JButton();
        EmployeeNum = new javax.swing.JTextField();

```

```

jButton1 = new javax.swing.JButton();
addEmployee = new javax.swing.JButton();
EmployeeFname = new javax.swing.JTextField();
EmployeeLname = new javax.swing.JTextField();
display = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jSeparator1 = new javax.swing.JSeparator();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

table.setBackground(new java.awt.Color(222, 251, 251));
table.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
table.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        },
    new String [] {
        "ID", "First Name", "Last Name"
    }
));
jScrollPane2.setViewportViewView(table);

deleteEmployee.setFont(new java.awt.Font("Tahoma", 1, 13)); //
NOI18N
deleteEmployee.setText("Delete");
deleteEmployee.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        deleteEmployeeActionPerformed(evt);
    }
});

jButton1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jButton1.setText("Home");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

addEmployee.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
addEmployee.setText("Add");
addEmployee.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        addEmployeeActionPerformed(evt);
    }
});

display.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
display.setText("Display");
display.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        displayActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel1.setText("ID :");

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel2.setText("First Name :");

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel3.setText("Last Name :");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel5.setForeground(new java.awt.Color(204, 0, 0));
jLabel5.setText("* To delete: you only need to enter the ID#");

jButton2.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jButton2.setText("Clear");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setText("Add Examples");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(jButton1,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(168, 168, 168))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
    .addContainerGap(33, Short.MAX_VALUE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAI
LING)

        .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAI
LING)

        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel2)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(EmployeeFname,
javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel1)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(EmployeeNum,
javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel3)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(EmployeeLname,
javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(159, 159, 159))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 372,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(38, 38, 38)))
    .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAI
LING)

        .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 251,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)

```

```

        .addGroup(layout.createSequentialGroup()
            .addGap(60, 60, 60)
            .addComponent(jLabel5))
        .addGroup(layout.createSequentialGroup()
            .addGap(35, 35, 35)
            .addComponent(jButton3,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(168, 168, 168)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(display,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(addEmployee,
javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(deleteEmployee,
javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(33, 33, 33)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(display)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(addEmployee)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(deleteEmployee)
        .addGap(18, 18, 18)
        .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel5)
        .addGap(29, 29, 29)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(EmployeeNum,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel1))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(EmployeeFname,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(EmployeeLname,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))
        .addGap(59, 59, 59)
        .addComponent(jButton1)
        .addGap(27, 27, 27))
    );

    pack();
} // </editor-fold> //GEN-END:initComponents

```

```

static EmployeeArrayList EID = new EmployeeArrayList();
Manager_Invoker invoker = new Manager_Invoker();
Employee e;
ErrorMessage em;
SuccessMessage sm;

private void deleteEmployeeActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_deleteEmployeeActionPerformed

    Manager_Commands deleteEmpl;

    int count = 0;

    for(int i = 0; i < EmployeeIDPage.EID.Size(); i++)
    {
        if
(EmployeeNum.getText().equalsIgnoreCase(EmployeeIDPage.EID.getEmployee(i).g
etID()))
        {
            count++;
            String id = EmployeeIDPage.EID.getEmployee(i).getID();

            deleteEmpl = new
removeEmployee(EmployeeIDPage.EID.getEmployee(i));
            invoker.execute_Manager_Commands(deleteEmpl);

            sm = new SuccessMessage("The employee with ID "+ id +" is
deleted successfully");
            sm.setVisible(true);

        }
        else {
            continue;
        }
    }

    if (count == 0) {

        em = new ErrorMessage("The employee ID number Does not exist,
please try again \n");
        em.setVisible(true);

    }
} //GEN-LAST:event_deleteEmployeeActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

```

```

{ //GEN-FIRST:event_jButtononActionPerformed
    dispose();
    new Home().setVisible(true);
} //GEN-LAST:event_jButtononActionPerformed

private void addEmployeeActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_addEmployeeActionPerformed

    Manager_Commands addEmpl;
    int count = 0;

    for(int i = 0; i < EmployeeIDPage.EID.Size(); i++)
    {
        if
(EmployeeNum.getText().equalsIgnoreCase(EmployeeIDPage.EID.getEmployee(i).g
etID()))
        {
            count ++;
            em = new ErrorMessage("The employee with ID " +
EmployeeNum.getText() + " already exists! \n");
            em.setVisible(true);
        }
        else
        {
            continue;
        }
    }

    if (count ==0)
    {
        if (!(checkID(EmployeeNum.getText())) ) {

            ErrorMessage EM = new ErrorMessage("only enter integers for
ID");

            EM.setVisible(true);

        }
        else if (EmployeeFname.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter the
employee first name.");
            EM.setVisible(true);

        }
        else if (EmployeeLname.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter the
employee last name.");
            EM.setVisible(true);

```



```

        } else {
            e = new Employee(EmployeeNum.getText(),
EmployeeFname.getText(), EmployeeLname.getText());

            addEmpl = new addEmployee(e);
            invoker.execute_Manager_Commands(addEmpl); // add employee
            sm = new SuccessMessage("Employee is added, and can access the
system now");
            sm.setVisible(true);
        }
    }

}

//GEN-LAST:event_addEmployeeActionPerformed

static boolean checkID(String number){
try
{
    Integer.parseInt(number);
}
catch (NumberFormatException e){
    return false;
}
return true;
}

private void displayActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_displayActionPerformed

    for (int i = 0; i < EmployeeIDPage.EID.Size(); i++) {

        model.insertRow(model.getRowCount(), new
Object[]{EmployeeIDPage.EID.getEmployee(i).getID(),
EmployeeIDPage.EID.getEmployee(i).getFame(),
EmployeeIDPage.EID.getEmployee(i).getLame()});

    }
}

//GEN-LAST:event_displayActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton2ActionPerformed
    model.setRowCount(0);
}
//GEN-LAST:event_jButton2ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton3ActionPerformed
    addID();
}
//GEN-LAST:event_jButton3ActionPerformed

```

```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTextField EmployeeFname;
private javax.swing.JTextField EmployeeLname;
private javax.swing.JTextField EmployeeNum;
private javax.swing.JButton addEmployee;
private javax.swing.JButton deleteEmployee;
private javax.swing.JButton display;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTable table;
// End of variables declaration//GEN-END:variables
}

```

ManagerPage_participant.java

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.util.ArrayList;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author esraaaljahdali
 */
public class ManagerPage_participant extends javax.swing.JFrame {

    /**

```

```

    * Creates new form ManagerPage_1
    */
    DefaultTableModel model;

    public ManagerPage_participant() {
        initComponents();
        setTitle("Manage Participant information");

        model = (DefaultTableModel) table.getModel();

        countryList.setSelectedIndex(-1);
    }

    public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
        WindowEvent.WINDOW_CLOSING);

        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
        );
    }

    /**
     * This method is called from within the constructor to initialize the
     form.
     * WARNING: Do NOT modify this code. The content of this method is
     always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
    BEGIN: initComponents
        private void initComponents() {

            displayList = new javax.swing.JButton();
            jScrollPane2 = new javax.swing.JScrollPane();
            table = new javax.swing.JTable();
            back = new javax.swing.JButton();
            addParticipant = new javax.swing.JButton();
            deleteParticipant = new javax.swing.JButton();
            txt1 = new javax.swing.JTextField();
            txt2 = new javax.swing.JTextField();
            txt3 = new javax.swing.JTextField();
            txt5 = new javax.swing.JTextField();
            txt6 = new javax.swing.JTextField();
            txt7 = new javax.swing.JTextField();
            txt8 = new javax.swing.JTextField();
            txt9 = new javax.swing.JTextField();
            txt11 = new javax.swing.JTextField();
            jLabel1 = new javax.swing.JLabel();

```

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
WorkshopMenuue = new javax.swing.JComboBox();
Clear = new javax.swing.JButton();
jButton1 = new javax.swing.JButton();
countryList = new javax.swing.JComboBox();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

displayList.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
displayList.setText("Display ");
displayList.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        displayListActionPerformed(evt);
    }
});

table.setBackground(new java.awt.Color(222, 251, 251));
table.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
table.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        },
    new String [] {
        "ID", "First Name", "Last Name", "Workshop", "Phone",
"Country", "City", "State", "Zip code"
    }
));
jScrollPane2.setViewportView(table);

back.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
back.setText("Back");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});

```

```

        addParticipant.setFont(new java.awt.Font("Tahoma", 1, 13)); //
NOI18N
        addParticipant.setText("Add");
        addParticipant.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                addParticipantActionPerformed(evt);
            }
        });

        deleteParticipant.setFont(new java.awt.Font("Tahoma", 1, 13)); //
NOI18N
        deleteParticipant.setText("Delete ");
        deleteParticipant.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                deleteParticipantActionPerformed(evt);
            }
        });

        txt1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txt1ActionPerformed(evt);
            }
        });

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(255, 102, 0));
        jLabel1.setText("Enter Participant Information:");

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(102, 0, 255));
        jLabel2.setText("ID :");

        jLabel3.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(102, 0, 255));
        jLabel3.setText("First name :");

        jLabel4.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(102, 0, 255));
        jLabel4.setText("Last name :");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(102, 0, 255));
        jLabel5.setText("Workshop :");

        jLabel6.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        jLabel6.setForeground(new java.awt.Color(102, 0, 255));
        jLabel6.setText("Phone:");

```

```

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel7.setForeground(new java.awt.Color(102, 0, 255));
jLabel7.setText("Address:");

jLabel8.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel8.setForeground(new java.awt.Color(102, 0, 255));
jLabel8.setText("Email :");

jLabel9.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel9.setForeground(new java.awt.Color(102, 0, 255));
jLabel9.setText("Country :");

jLabel10.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel10.setForeground(new java.awt.Color(102, 0, 255));
jLabel10.setText("City :");

jLabel11.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel11.setForeground(new java.awt.Color(102, 0, 255));
jLabel11.setText("State :");

jLabel12.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel12.setForeground(new java.awt.Color(102, 0, 255));
jLabel12.setText("Zip :");

jLabel14.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLabel14.setForeground(new java.awt.Color(255, 51, 0));
jLabel14.setText("* To delete, you only need to enter ID#");

WorkshopMenu.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Computer Graphics", "Computer Security", "Artificial
Intelligence", "Software Studio", "Feedback Systems" }));

Clear.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
Clear.setText("Clear");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});

jButton1.setText("Add Examples");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

countryList.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "United States", "Albania", "Algeria", "Andorra", "Angola",
"Antigua and Barbuda", "Argentina", "Afghanistan", "Armenia", "Aruba",

```

```
countryList.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        countryListActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout layout = new  
javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
  
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup()  
        .addGap(54, 54, 54)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup()
```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup()
                    .addGap(171, 171, 171)
                    .addComponent(displayList,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(Clear,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(addParticipant,
javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(deleteParticipant,
javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(55, 55, 55)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jLabel1)
                .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 701,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

                .addGap(73, 73, 73)
                .addComponent(jLabel2))
                .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel3,
javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                .addComponent(txt3)
                .addComponent(txt2,

```



```

javax.swing.GroupLayout.Alignment.TRAILING)
                                .addComponent(txt1,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(jLabel5)
                                .addComponent(jLabel7,
javax.swing.GroupLayout.Alignment.TRAILING)
                                .addComponent(jLabel6,
javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(txt5,
javax.swing.GroupLayout.DEFAULT_SIZE, 114, Short.MAX_VALUE)
                                .addComponent(txt6))
                                .addGap(77, 77, 77)
                                .addComponent(jLabel10))

.addGroup(layout.createSequentialGroup())
                                .addGap(167, 167, 167)
                                .addComponent(jLabel9))

.addGroup(layout.createSequentialGroup())
                                .addComponent(WorkshopMenue,
javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(28, 28, 28)
                                .addComponent(jLabel8))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(countryList, 0, 97,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

```

```

        .addComponent(txt7,
javax.swing.GroupLayout.DEFAULT_SIZE, 97, Short.MAX_VALUE)
        .addComponent(txt8)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel11))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(60, 60, 60)
        .addComponent(jLabel12)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(txt9,
javax.swing.GroupLayout.DEFAULT_SIZE, 82, Short.MAX_VALUE)
        .addComponent(txt11)))
        .addGroup(layout.createSequentialGroup()
            .addGap(41, 41, 41)
            .addComponent(jLabel14,
javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(51, 66, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(back,
javax.swing.GroupLayout.PREFERRED_SIZE, 83,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(56, 56, 56))
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 791,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createSequentialGroup()
            .addGap(6, 6, 6)
            .addComponent(jButton1)))
        .addGap(0, 0, Short.MAX_VALUE)))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
            .addGap(34, 34, 34)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup())

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 4,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup())
                .addGap(47, 47, 47)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(displayList)
            .addComponent(addParticipant)
            .addComponent(Clear)
            .addComponent(deleteParticipant))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20,
Short.MAX_VALUE))
            .addGap(18, 18, 18)
            .addComponent(jLabel14,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 7,
Short.MAX_VALUE)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(txt1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txt7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(txt9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel12)
        .addComponent(jLabel15)
        .addComponent(jLabel18)
        .addComponent(jLabel11)
        .addComponent(WorkshopMenue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(2, 2, 2)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txt2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txt5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txt11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel13)
        .addComponent(jLabel16)
        .addComponent(jLabel19)
        .addComponent(jLabel112)
        .addComponent(countryList,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(2, 2, 2)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txt3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txt6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txt8,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel4)
    .addComponent(jLabel7)
    .addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 59,
Short.MAX_VALUE)
    .addComponent(back, javax.swing.GroupLayout.PREFERRED_SIZE,
37, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(26, 26, 26))
);

pack();
} // </editor-fold> // GEN-END: initComponents

static ParticipantArrayList as = new ParticipantArrayList();

Manager_Invoker invoker = new Manager_Invoker();

ErrorMessage em;
SuccessMessage sm;

public void addParticipantExamples() {

    // add participants examples

    as.addParticipant(new Participant(235, "Esraa", "Aljahdali",
"Computer Graphics", "5654343678", "", "esraa@gmail.com", "Minneapolis",
"MN", "United States", "55646", 0));
    as.addParticipant(new Participant(543, "Alaa", "Hofman", "Software
Studio", "8767659845", "", "alaa@gmail.com", "St. Paul", "MN", "United
States", "56444", 0));
    as.addParticipant(new Participant(245, "Jane", "peter", "Computer
Graphics", "6545676909", "", "jane@hotmail.com", "New York", "NY", "United
States", "54443", 0));
    as.addParticipant(new Participant(776, "Kevin", "Ron", "Computer
Security", "7663421125", "", "kevin@yahoo.com", "St. Cloud", "MN", "United
States", "55434", 0));
    as.addParticipant(new Participant(555, "Sara", "Micheal", "Feedback
Systems", "6544453219", "", "sara@yahoo.com", "Springfield", "IL", "United
States", "55123", 0));
    as.addParticipant(new Participant(112, "Bander", "Jahdali",
"Artificial Intelligence", "9876576565", "", "bander@hotmail.sa.com",
"Orlando", "FL", "United States", "66543", 0));
    as.addParticipant(new Participant(854, "Hana", "Joy", "Feedback
Systems", "4546787657", "", "hana@gmail.com", "Phoenix", "AZ", "United
States", "56675", 0));
    as.addParticipant(new Participant(654, "Jawharah", "Jahdali",
"Artificial Intelligence", "6540098906", "", "jawhara@sayahoo.com",

```

```

"Orlando", "FL", "United States", "66543", 0));
    as.addParticipant(new Participant(112, "Martin", "King",
"Artificial Intelligence", "8767655444", "", "martin@MKgmail.com",
"Juneau", "AK", "United States", "87870", 0));

}

    private void displayListActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_displayListActionPerformed

        for (int i = 0; i < as.getSize(); i++) {
            model.insertRow(model.getRowCount(), new
Object[]{as.getParticipant(i).getID(), as.getParticipant(i).getFame(),
as.getParticipant(i).getLame(), as.getParticipant(i).getWorkshopType(),
as.getParticipant(i).getPhone(), as.getParticipant(i).getCountry(),
as.getParticipant(i).getCity(), as.getParticipant(i).getState(),
as.getParticipant(i).getZip()});
        }

    //GEN-LAST:event_displayListActionPerformed

    private void backActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_backActionPerformed
        dispose();

        new Home().setVisible(true);
    //GEN-LAST:event_backActionPerformed

    private void addParticipantActionPerformed(java.awt.event.ActionEvent
evt) {
    //GEN-FIRST:event_addParticipantActionPerformed

        Manager_Commands addCus;
        Participant c;
        String WorkshopType;
        int count = 0;
        int index;

        for (int i = 0; i < as.getSize(); i++) {
            if
(txt1.getText().equalsIgnoreCase(Integer.toString(as.getParticipant(i).getI
D())))) {
                count++;
                em = new ErrorMessage(" Participant with ID " +
txt1.getText() + " already exists! \n");
                em.setVisible(true);
            } else {
                continue;
            }
        }
    }
}

```

```

    if (count == 0) {
        if (WorkshopMenu.getSelectedIndex() == 0) {

            if (Report.ComputerGraphics.Count == 0) {
                ErrorMessage EM = new ErrorMessage("Computer Graphics
workshop is full, select another workshop");
                EM.setVisible(true);
            } else {

                if (!(checkID(txt1.getText()))) {

                    ErrorMessage EM = new ErrorMessage("only enter
integers for ID");
                    EM.setVisible(true);

                } else if (txt2.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Please enter
the participant first name.");
                    EM.setVisible(true);

                } else if (txt3.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Please enter
the participant last name.");
                    EM.setVisible(true);

                } else if (txt5.getText().isEmpty() ||
(txt5.getText().length() != 10) || !(checkID(txt5.getText()))) {

                    ErrorMessage EM = new ErrorMessage("Please enter a
valid a phone number with 10 digits");
                    EM.setVisible(true);

                } else if (txt6.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Please enter
the participant address");
                    EM.setVisible(true);

                } else if (txt7.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Please enter
the participant Email address");
                    EM.setVisible(true);

                } else if (txt8.getText().isEmpty()) {

                    ErrorMessage EM = new ErrorMessage("Please enter

```

```

the participant city");
        EM.setVisible(true);

        } else if (txt9.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant state");
            EM.setVisible(true);

        } else if (countryList.getSelectedIndex() == -1) {

            ErrorMessage EM = new ErrorMessage("Please select
the participant country");
            EM.setVisible(true);

        } else if (txt11.getText().isEmpty() ||
(txt11.getText().length() != 5) || !(checkID(txt11.getText())) {

            ErrorMessage EM = new ErrorMessage("Please enter
the zip code with 5 digits");
            EM.setVisible(true);

        } else {
            WorkshopType = "Computer Graphics";
            index = countryList.getSelectedIndex();

            c = new
Participant(Integer.parseInt(txt1.getText()), txt2.getText(),
txt3.getText(), WorkshopType, txt5.getText(), txt6.getText(),
txt7.getText(), txt8.getText(), txt9.getText(),
countryList.getSelectedItem().toString(), txt11.getText(), index);
            addCus = new AddParticipant(c);
            invoker.execute_Manager_Commands(addCus); // add
participant

            sm = new SuccessMessage("Participant is added
successfully");
            sm.setVisible(true);
        }
    }

    } else if (WorkshopMenu.getSelectedIndex() == 1) {
        if (Report.ComputerSecurity.Count == 0) {
            ErrorMessage EM = new ErrorMessage("Computer Security
workshop is full, select another workshop");
            EM.setVisible(true);
        } else {
            if (!(checkID(txt1.getText()))) {

                ErrorMessage EM = new ErrorMessage("only enter

```



```

integers for ID");
        EM.setVisible(true);
    } else if (txt2.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant first name.");
        EM.setVisible(true);
    } else if (txt3.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant last name.");
        EM.setVisible(true);
    } else if (txt5.getText().isEmpty() ||
(txt5.getText().length() != 10) || !(checkID(txt5.getText())) {
        ErrorMessage EM = new ErrorMessage("Please enter a
valid a phone number with 10 digits");
        EM.setVisible(true);
    } else if (txt6.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant address");
        EM.setVisible(true);
    } else if (txt7.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant Email address");
        EM.setVisible(true);
    } else if (txt8.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant city");
        EM.setVisible(true);
    } else if (txt9.getText().isEmpty()) {
        ErrorMessage EM = new ErrorMessage("Please enter
the participant state");
        EM.setVisible(true);
    } else if (countryList.getSelectedIndex() == -1) {
        ErrorMessage EM = new ErrorMessage("Please select
the participant country");
    }

```

```

        EM.setVisible(true);

        } else if (txt11.getText().isEmpty() ||
(txt11.getText().length() != 5) || !(checkID(txt11.getText())) {

            ErrorMessage EM = new ErrorMessage("Please enter
the zip code with 5 digits");
            EM.setVisible(true);

        } else {
            WorkshopType = "Computer Security";
            index = countryList.getSelectedIndex();

            c = new
Participant(Integer.parseInt(txt1.getText()), txt2.getText(),
txt3.getText(), WorkshopType, txt5.getText(), txt6.getText(),
txt7.getText(), txt8.getText(), txt9.getText(),
countryList.getSelectedItem().toString(), txt11.getText(), index);
            addCus = new AddParticipant(c);
            invoker.execute_Manager_Commands(addCus); // add
participant

            sm = new SuccessMessage("Participant is added
successfully");
            sm.setVisible(true);
        }

    }
} else if (WorkshopMenu.getSelectedIndex() == 2) {

    if (Report.ArtificialIntelligence.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Artificial
Intelligence workshop is full, select another workshop");
        EM.setVisible(true);
    } else {

        if (!(checkID(txt1.getText())) {

            ErrorMessage EM = new ErrorMessage("only enter
integers for ID");
            EM.setVisible(true);

        } else if (txt2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant first name.");
            EM.setVisible(true);

        } else if (txt3.getText().isEmpty()) {

```

```

        ErrorMessage EM = new ErrorMessage("Please enter
the participant last name.");
        EM.setVisible(true);

        } else if (txt5.getText().isEmpty() ||
(txt5.getText().length() != 10) || !(checkID(txt5.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter a
valid a phone number with 10 digits");
        EM.setVisible(true);

        } else if (txt6.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant address");
        EM.setVisible(true);

        } else if (txt7.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant Email address");
        EM.setVisible(true);

        } else if (txt8.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant city");
        EM.setVisible(true);

        } else if (txt9.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant state");
        EM.setVisible(true);

        } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select
the participant country");
        EM.setVisible(true);

        } else if (txt11.getText().isEmpty() ||
(txt11.getText().length() != 5) || !(checkID(txt11.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter
the zip code with 5 digits");
        EM.setVisible(true);

        } else {
            WorkshopType = "Artificial Intelligence";

```

```

        index = countryList.getSelectedIndex();

        c = new
Participant(Integer.parseInt(txt1.getText()), txt2.getText(),
txt3.getText(), WorkshopType, txt5.getText(), txt6.getText(),
txt7.getText(), txt8.getText(), txt9.getText(),
countryList.getSelectedItem().toString(), txt11.getText(), index);
        addCus = new AddParticipant(c);
        invoker.execute_Manager_Commands(addCus); // add
participant

        sm = new SuccessMessage("Participant is added
successfully");
        sm.setVisible(true);
    }
}
} else if (WorkshopMenue.getSelectedIndex() == 3) {
    if (Report.SoftwareStudio.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Software Studio
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        if (!(checkID(txt1.getText()))) {

            ErrorMessage EM = new ErrorMessage("only enter
integers for ID");
            EM.setVisible(true);

        } else if (txt2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant first name.");
            EM.setVisible(true);

        } else if (txt3.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant last name.");
            EM.setVisible(true);

        } else if (txt5.getText().isEmpty() ||
(txt5.getText().length() != 10) || !(checkID(txt5.getText()))) {

            ErrorMessage EM = new ErrorMessage("Please enter a
valid a phone number with 10 digits");
            EM.setVisible(true);

        } else if (txt6.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter

```

```

the participant address");
        EM.setVisible(true);

    } else if (txt7.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant Email address");
        EM.setVisible(true);

    } else if (txt8.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant city");
        EM.setVisible(true);

    } else if (txt9.getText().isEmpty()) {

        ErrorMessage EM = new ErrorMessage("Please enter
the participant state");
        EM.setVisible(true);

    } else if (countryList.getSelectedIndex() == -1) {

        ErrorMessage EM = new ErrorMessage("Please select
the participant country");
        EM.setVisible(true);

    } else if (txt11.getText().isEmpty() ||
(txt11.getText().length() != 5) || !(checkID(txt11.getText())) {

        ErrorMessage EM = new ErrorMessage("Please enter
the zip code with 5 digits");
        EM.setVisible(true);

    } else {
        WorkshopType = "Software Studio";
        index = countryList.getSelectedIndex();

        c = new
Participant(Integer.parseInt(txt1.getText()), txt2.getText(),
txt3.getText(), WorkshopType, txt5.getText(), txt6.getText(),
txt7.getText(), txt8.getText(), txt9.getText(),
countryList.getSelectedItem().toString(), txt11.getText(), index);
        addCus = new AddParticipant(c);
        invoker.execute_Manager_Commands(addCus); // add
participant

        sm = new SuccessMessage("Participant is added
successfully");
        sm.setVisible(true);

```

```

    }
}
} else if (WorkshopMenu.getSelectedIndex() == 4) {
    if (Report.FeedbackSystems.Count == 0) {
        ErrorMessage EM = new ErrorMessage("Feedback Systems
workshop is full, select another workshop");
        EM.setVisible(true);
    } else {
        if (!(checkID(txt1.getText()))) {

            ErrorMessage EM = new ErrorMessage("only enter
integers for ID");
            EM.setVisible(true);

        } else if (txt2.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant first name.");
            EM.setVisible(true);

        } else if (txt3.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant last name.");
            EM.setVisible(true);

        } else if (txt5.getText().isEmpty() ||
(txt5.getText().length() != 10) || !(checkID(txt5.getText()))) {

            ErrorMessage EM = new ErrorMessage("Please enter a
valid a phone number with 10 digits");
            EM.setVisible(true);

        } else if (txt6.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant address");
            EM.setVisible(true);

        } else if (txt7.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant Email address");
            EM.setVisible(true);

        } else if (txt8.getText().isEmpty()) {

            ErrorMessage EM = new ErrorMessage("Please enter
the participant city");
            EM.setVisible(true);

```



```

        int count = 0;

        for (int i = 0; i < as.getSize(); i++) {
            if
(txt1.getText().equalsIgnoreCase(Integer.toString(as.getParticipant(i).getI
D())))) {
                count++;
                int num = as.getParticipant(i).getID();

                deleteCus = new RemoveParticipant(as.getParticipant(i));
                invoker.execute_Manager_Commands(deleteCus);

                sm = new SuccessMessage("Participant with ID# " + num + "
is deleted successfully");
                sm.setVisible(true);

            } else {
                continue;
            }
        }

        if (count == 0) {
            em = new ErrorMessage(" Participant with ID " + txt1.getText()
+ " does not exist! \n");
            em.setVisible(true);
        }

    } //GEN-LAST:event_deleteParticipantActionPerformed

    static boolean checkID(String number) {
        try {
            Integer.parseInt(number);
        } catch (NumberFormatException e) {
            return false;
        }
        return true;
    }

    private void ClearActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_ClearActionPerformed
        model.setRowCount(0);
    } //GEN-LAST:event_ClearActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton1ActionPerformed
        addParticipantExamples();
    } //GEN-LAST:event_jButton1ActionPerformed

    private void txt1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_txt1ActionPerformed
        // TODO add your handling code here:

```



```

    }//GEN-LAST:event_txt1ActionPerformed

    private void countryListActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_countryListActionPerformed

    }//GEN-LAST:event_countryListActionPerformed


    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton Clear;
    private javax.swing.JComboBox WorkshopMenu;
    private javax.swing.JButton addParticipant;
    private javax.swing.JButton back;
    private javax.swing.JComboBox countryList;
    private javax.swing.JButton deleteParticipant;
    private javax.swing.JButton displayList;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JTable table;
    private javax.swing.JTextField txt1;
    private javax.swing.JTextField txt11;
    private javax.swing.JTextField txt2;
    private javax.swing.JTextField txt3;
    private javax.swing.JTextField txt5;
    private javax.swing.JTextField txt6;
    private javax.swing.JTextField txt7;
    private javax.swing.JTextField txt8;
    private javax.swing.JTextField txt9;
    // End of variables declaration//GEN-END:variables
}

```

addEmployee.java

```

package Esraa_SP_Project;

public class addEmployee implements Manager_Commands {

    private Employee employee;

    public addEmployee(Employee employee) {

        this.employee = employee;
    }

    public void execute()
    {

        EmployeeIDPage.EID.addEmployee(employee);

    }

}

```

addParticipant.java

```

package Esraa_SP_Project;

import java.util.ArrayList;

public class AddParticipant implements Manager_Commands{

    private Participant p;

    public AddParticipant(Participant participant) {
        this.p = participant;
    }

    public void execute() {

        ManagerPage_participant.as.addParticipant(p);

    }

}

```

```

        if ("Computer Graphics".equals(p.getWorkshopType()))
        {
            Report.ComputerGraphics.decrementCount();
        }
        else if ("Computer Security".equals(p.getWorkshopType()))
        {
            Report.ComputerSecurity.decrementCount();
        }
        else if ("Artificial Intelligence".equals(p.getWorkshopType()))
        {
            Report.ArtificialIntelligence.decrementCount();
        }
        else if ("Software Studio".equals(p.getWorkshopType()))
        {
            Report.SoftwareStudio.incrementCount();
        }
        else if ("Feedback Systems".equals(p.getWorkshopType()))
        {
            Report.FeedbackSystems.decrementCount();
        }
    }
}

```

removeEmployee.java

```

package Esraa_SP_Project;

public class removeEmployee implements Manager_Commands {

    private Employee employee;

    public removeEmployee(Employee employee) {

        this.employee = employee;
    }

    public void execute()
    {

```

```

        EmployeeIDPage.EID.removeEmployee(employee);
    }
}

```

RemoveParticipant.java

```

package Esraa_SP_Project;

public class RemoveParticipant implements Manager_Commands {
    private Participant participant;

    public RemoveParticipant(Participant participant)
    {
        this.participant = participant;
    }

    public void execute()
    {
        ManagerPage_participant.as.removeParticipant(participant);

        if ("Computer Graphics".equals(participant.getWorkshopType()))
        {
            Report.ComputerGraphics.incrementCount();
        }
        else if ("Computer Security".equals(participant.getWorkshopType()))
        {
            Report.ComputerSecurity.incrementCount();
        }
        else if ("Artificial
Intelligence".equals(participant.getWorkshopType()))
        {
            Report.ArtificialIntelligence.incrementCount();
        }
        else if ("Software Studio".equals(participant.getWorkshopType()))
        {

```

```

        Report.SoftwareStudio.incrementCount();
    }
    else if ("Feedback Systems".equals(participant.getWorkshopType()))
    {
        Report.FeedbackSystems.incrementCount();
    }
}
}
}

```

AddPlace.java

```

package Esraa_SP_Project;

public class AddPlace implements Employee_Commands {

    private Workshops workshop;

    public AddPlace(Workshops workshop) {

        this.workshop = workshop;
    }

    public void execute()
    {
        workshop.incrementCount();
    }

}

```

RemovePlace.java

```

package Esraa_SP_Project;

public class RemovePlace implements Employee_Commands {

```

```
private Workshops workshop;

public RemovePlace(Workshops workshop) {

    this.workshop = workshop;
}

public void execute()
{
    workshop.decrementCount();
}
}
```

changeFee.java

```
package Esraa_SP_Project;

public class changeFee implements Employee_Commands {

    private Workshops workshop;
    int fee;

    public changeFee(Workshops workshop, int fee) {
        this.workshop = workshop;
        this.fee = fee;
    }

    public void execute()
    {
        workshop.setFee(fee);
    }
}
```

changeHours.java

```
package Esraa_SP_Project;
```

```

public class changeHours implements Employee_Commands {

    private Workshops workshop;

    int period;

    public changeHours(Workshops workshop, int period) {
        this.workshop = workshop;
        this.period = period;
    }

    public void execute()
    {

        workshop.setPeriod(period);

    }

}

```

ArtificialIntelligence.java

```

package Esraa_SP_Project;

public class ArtificialIntelligence{

    int ID;
    String type;
    int fee;
    int Period;
    int Count;
    int numOfPlaces;

    public ArtificialIntelligence()
    {
        ID = 3;
        type="Artificial Intelligence";
        fee = 1000;
        Period = 37;
        Count=10;
        numOfPlaces = 10;
    }
}

```

```

    }
    public int geNumOfPlaces()
    {
        return numOfPlaces;
    }
    public void setNumOfPlaces()
    {
        this.numOfPlaces = numOfPlaces;
    }
}

```

ComputerGraphics.java

```

package Esraa_SP_Project;

public class ComputerGraphics {

    int ID;
    String type;
    int Fee;
    int Period;
    int Count;
    int numOfPlaces;

    public ComputerGraphics()
    {
        ID = 1;
        type = "Computer Graphics";
        Fee = 200;
        Period = 21;
        Count = 10;
        numOfPlaces = 10;
    }

    public int geNumOfPlaces()
    {
        return numOfPlaces;
    }

    public void setNumOfPlaces(int NumOfPlaces)

```



```

{
    this.numOfPlaces = NumOfPlaces;
}

```

ComputerSecurity.java

```

package Esraa_SP_Project;

public class ComputerSecurity {
    int ID;
    String type;
    int fee;
    int Period;
    int Count;
    int numOfPlaces;

    public ComputerSecurity()
    {
        ID = 2;
        type = "Computer Security";
        fee = 500;
        Period = 19;
        Count = 10;
        numOfPlaces = 10;
    }

    public ComputerSecurity(int numOfPlaces){

        numOfPlaces = numOfPlaces;

    }
    public int geNumOfPlaces() {

        return numOfPlaces;
    }
    public void setNumOfPlaces() {

        this.numOfPlaces = numOfPlaces;
    }
}

```

SoftwareStudio.java

```
package Esraa_SP_Project;

public class SoftwareStudio {

    int ID;
    String type;
    int fee;
    int Period;
    int Count;
    int numOfPlaces;

    public SoftwareStudio()
    {
        ID = 4;
        type = "Software Studio";
        fee = 200;
        Period = 28;
        Count = 10;
        numOfPlaces = 10;
    }

    public int geNumOfPlaces()
    {
        return numOfPlaces;
    }

    public void setNumOfPlaces()
    {
        this.numOfPlaces = numOfPlaces;
    }
}
```

FeedbackSystems.java

```
package Esraa_SP_Project;

public class FeedbackSystems {

    int ID;
    String type;
```

```

int Fee;
int Period;
int Count;
int numOfPlaces;

public FeedbackSystems()
{
    ID = 5;
    type = "Computer Graphics";
    Fee = 450;
    Period = 17;
    Count = 10;
    numOfPlaces = 10;
}

public int geNumOfPlaces()
{
    return numOfPlaces;
}

public void setNumOfPlaces(int NumOfPlaces)
{
    this.numOfPlaces = NumOfPlaces;
}
}

```

Context_PaymentStrategy.java

```

package Esraa_SP_Project;

public class Context_PaymentStrategy {
    private PaymentStrategy strategy;
    //Constructor
    public Context_PaymentStrategy(PaymentStrategy strategy1)
    {
        this.strategy = strategy1;
    }
    //execute PaymentStrategy
    public String executeStrategy(int amount)
    {
        return strategy.pay(amount);
    }
}

```

PaymentStrategy.java

```
package Esraa_SP_Project;

public interface PaymentStrategy {
    String pay(int amount);
}
```

cash.java

```
public class Cash implements PaymentStrategy{

    public String pay(int amount)
    {
        String string = amount + "$ fee should be paid within 3 days in office";
        return string;
    }
}
```

Visa.java

```
public class Visa implements PaymentStrategy{

    private String name;
    private String cardNumber;
    private String cvv;
    private String dateOfExpiry;

    public Visa(String nm, String ccNum, String cvv, String expiryDate)
    {
        this.name=nm;
        this.cardNumber=ccNum;
        this.cvv=cvv;
        this.dateOfExpiry=expiryDate;
    }

    public String pay(int amount)
```

```

    {
        String string = amount + " will be paid with visa card";
        return string;
    }
}

```

Paypal.java

```

package Esraa_SP_Project;

public class Paypal implements PaymentStrategy{

    private String emailId;
    private String password;

    public Paypal(String email, String pwd)
    {
        this.emailId = email;
        this.password = pwd;
    }

    public String pay(int amount)
    {
        String string = amount + "$ fee will be paid using Paypal";
        return string;
    }

}

```

Context_RegisterStrategy.java

```

package Esraa_SP_Project;

class Context_RegisterStrategy {
    private registerStrategy registerS;
}

```

```

    public Context_RegisterStrategy(registerStrategy registerS) {
        this.registerS = registerS;
    }

    public String execute() {
        return registerS.register();
    }
}

```

registerStrategy.java

```

package Esraa_SP_Project;

interface registerStrategy
{
    String register();
}

```

EmailRegistration.java

```

package Esraa_SP_Project;

class EmailRegistration implements registerStrategy {

    public String register() {
        return "<html>Dear participant, please send your information<br>to  
our Email address: CSWorkshop@register.com</html> " ;
    }
}

```

TelephoneRegistration.java

```

package Esraa_SP_Project;

class TelephoneRegistration implements registerStrategy {

```

```

        public String register() {
            return "<html>Dear participant, please contact
Registration<br>Services (320) 567-5674<br>(M-F 8am - 4:30pm, Sat. & Sun.
8am - 2pm)<br>or CS Workshop office 320-432-1234.</html> " ;
        }
    }
}

```

OfficeRegistration.java

```

package Esraa_SP_Project;

class OfficeRegistration implements registerStrategy
{
    public String register() {
        return "<html>Dear participant, please visit our office address<br>CS
Workshops registration office<br>720 4th Avenue South<br>St. Cloud, MN
56301-4498 </html>" ;
    }
}

```

OnlineRegistration.java

```

package Esraa_SP_Project;

class OnlineRegistration implements registerStrategy {
    public String register() {
        return "You choose to register online" ;
    }
}

```

Employee_Commands.java

```
package Esraa_SP_Project;

public interface Employee_Commands {

    public void execute();

}
```

Employee_Invoker.java

```
package Esraa_SP_Project;

import java.util.ArrayList;
import java.util.List;

class Employee_Invoker {

    public Employee_Invoker() {
    }

    void execute_Employee_Commands(Employee_Commands command) {

        command.execute();

    }

}
```

EmployeeIDPage.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Esraa_SP_Project;

import java.awt.Toolkit;
```



```

import java.awt.event.WindowEvent;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author esraa Aljahdali
 */
public class EmployeeIDPage extends javax.swing.JFrame {

    /**
     * Creates new form EmployeePage
     */

    public EmployeeIDPage() {
        initComponents();
        setTitle("Employee ID page");
        //addID();
    }

    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }
    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel1 = new javax.swing.JLayeredPane();
        jLabel1 = new javax.swing.JLabel();
        EmployeeNum = new javax.swing.JTextField();
        Enter = new javax.swing.JButton();
        Back = new javax.swing.JButton();
        ErrorMessage = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Adobe Caslon Pro", 1, 15)); //
NOI18N
jLabel1.setText("Enter your ID#");
jLayeredPanel.add(jLabel1);
jLabel1.setBounds(170, 200, 100, 20);

EmployeeNum.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EmployeeNumActionPerformed(evt);
    }
});
jLayeredPanel.add(EmployeeNum);
EmployeeNum.setBounds(130, 220, 170, 34);

Enter.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
Enter.setText("Enter");
Enter.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EnterActionPerformed(evt);
    }
});
jLayeredPanel.add(Enter);
Enter.setBounds(180, 310, 80, 30);

Back.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
Back.setText("Back");
Back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BackActionPerformed(evt);
    }
});
jLayeredPanel.add(Back);
Back.setBounds(30, 390, 77, 30);

ErrorMessage.setFont(new java.awt.Font("Adobe Caslon Pro", 1, 14));
// NOI18N
ErrorMessage.setForeground(new java.awt.Color(255, 0, 0));
jLayeredPanel.add(ErrorMessage);
ErrorMessage.setBounds(120, 260, 211, 30);

jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/employeePic
.gif"))); // NOI18N
jLayeredPanel.add(jLabel2);
jLabel2.setBounds(-10, -20, 470, 530);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

```

```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 447,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap())
            );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 447,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            );

        pack();
    } // </editor-fold> // GEN-END: initComponents

    static EmployeeArrayList EID = new EmployeeArrayList();

    private void BackActionPerformed(java.awt.event.ActionEvent evt)
    { // GEN-FIRST:event_BackActionPerformed

        close();
        new Home().setVisible(true);
    } // GEN-LAST:event_BackActionPerformed

    private void EnterActionPerformed(java.awt.event.ActionEvent evt)
    { // GEN-FIRST:event_EnterActionPerformed

        for(int i = 0; i < EID.Size(); i++)
        {
            if
(EmployeeNum.getText().equalsIgnoreCase(EmployeeIDPage.EID.getEmployee(i).g
etID()))
            {
                close();
                new EmployeePage_1().setVisible(true);
            }
            else

```

```

        continue;
    }
    ErrorMessage.setText("ID does not exist, Please try again!");
} //GEN-LAST:event_EnterActionPerformed

private void EmployeeNumActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_EmployeeNumActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_EmployeeNumActionPerformed

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton Back;
private javax.swing.JTextField EmployeeNum;
private javax.swing.JButton Enter;
private javax.swing.JLabel ErrorMessage;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLayeredPane jLayeredPanel;
// End of variables declaration //GEN-END:variables
}

```

Employee.java

```

package Esraa_SP_Project;

public class Employee {

    String ID;
    String Fname;
    String Lname;

    public Employee(String id, String fname, String lname)
    {
        ID = id;
        Fname = fname;
        Lname = lname;
    }

    public void setID(String id)
    {

```

```

        ID = id;
    }
    public String getID()
    {
        return ID;
    }

    public void setFname(String fname)
    {
        Fname = fname;
    }
    public String getFame()
    {
        return Fname;
    }

    public void setLname(String lname)
    {
        Lname = lname;
    }
    public String getLame()
    {
        return Lname;
    }
}

```

EmployeeArrayList.java

```

package Esraa_SP_Project;

import java.util.ArrayList;

public class EmployeeArrayList {

    ArrayList employee = new ArrayList();

    public void addEmployee(Employee e)
    {
        employee.add(e);
    }
}

```

```

    public void removeEmployee(Employee e){
        employee.remove(e);
    }

    public Employee getEmployee(int index){
        return (Employee)employee.get(index);
    }

    public int Size()
    {
        return employee.size();
    }
}

```

EmployeePage_1.java

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.util.ArrayList;
import java.util.List;
//import java.util.Observable;
/**
 *
 * @author Esraa Aljahdali
 */
public class EmployeePage_1 extends javax.swing.JFrame {

    /**
     * Creates new form EmployeePage
     */

    public EmployeePage_1() {
        initComponents();

        enterFee.setVisible(false);
    }
}

```

```

        addPlace.setVisible(false);
        removePlace.setVisible(false);
        changeFee.setVisible(false);
        updateFee.setVisible(false);
        text1.setVisible(false);
        Menue.setVisible(false);
        updatePeriod.setVisible(false);
        enterPeriod.setVisible(false);
        changePeriod.setVisible(false);

    }

    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN:initComponents
    private void initComponents() {

        jLayeredPanel = new javax.swing.JLayeredPane();
        Manage = new javax.swing.JButton();
        Back = new javax.swing.JButton();
        DisplayReports = new javax.swing.JButton();
        Back1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        removePlace = new javax.swing.JButton();
        addPlace = new javax.swing.JButton();
        changeFee = new javax.swing.JButton();
        Menue = new javax.swing.JComboBox();
        text1 = new javax.swing.JLabel();
        message = new javax.swing.JLabel();
        enterFee = new javax.swing.JTextField();
        updateFee = new javax.swing.JButton();
        changePeriod = new javax.swing.JButton();
        enterPeriod = new javax.swing.JTextField();

```

```

updatePeriod = new javax.swing.JButton();
jSeparator1 = new javax.swing.JSeparator();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

Manage.setBackground(new java.awt.Color(255, 255, 255));
Manage.setText("Manage Workshops");
Manage.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ManageWorkshopActionPerformed(evt);
    }
});
jLayeredPanel1.add(Manage);
Manage.setBounds(20, 110, 160, 40);

Back.setText("Back");
Back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BackActionPerformed(evt);
    }
});
jLayeredPanel1.add(Back);
Back.setBounds(20, 210, 160, 40);

DisplayReports.setBackground(new java.awt.Color(219, 249, 249));
DisplayReports.setText("Display Reports");
DisplayReports.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        DisplayReportsActionPerformed(evt);
    }
});
jLayeredPanel1.add(DisplayReports);
DisplayReports.setBounds(20, 160, 160, 40);

Back1.setText("Home");
Back1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        HomeActionPerformed(evt);
    }
});
jLayeredPanel1.add(Back1);
Back1.setBounds(20, 260, 160, 40);

jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/EMTool.gif"
))); // NOI18N
jLayeredPanel1.add(jLabel2);
jLabel2.setBounds(0, -20, 630, 430);

```



```

removePlace.setBackground(new java.awt.Color(255, 255, 255));
removePlace.setText("Remove Place");
removePlace.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        RemoveSeatsActionPerformed(evt);
    }
});

addPlace.setBackground(new java.awt.Color(255, 255, 255));
addPlace.setText("Add Place");
addPlace.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        AddSeatsActionPerformed(evt);
    }
});

changeFee.setBackground(new java.awt.Color(255, 255, 255));
changeFee.setText("Change Fee");
changeFee.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        changeFeeHomeActionPerformed(evt);
    }
});

Menue.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
    "Computer Graphics", "Computer Security", "Artificial Intelligence",
    "Software Studio", "Feedback Systems" }));
Menue.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        MenueActionPerformed(evt);
    }
});

text1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
text1.setText("Please make a workshop selection to manage :");

message.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
message.setForeground(new java.awt.Color(220, 50, 7));

enterFee.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        enterFeeActionPerformed(evt);
    }
});

updateFee.setBackground(new java.awt.Color(253, 234, 212));
updateFee.setText("Update Fee");
updateFee.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        updateFeeActionPerformed(evt);
    }
});

changePeriod.setBackground(new java.awt.Color(255, 255, 255));
changePeriod.setText("Change Period");
changePeriod.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        changePeriodHomeActionPerformed(evt);
    }
});

enterPeriod.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        enterPeriodActionPerformed(evt);
    }
});

updatePeriod.setBackground(new java.awt.Color(251, 237, 208));
updatePeriod.setText("Update Period");
updatePeriod.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updatePeriodActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(41, 41, 41)
                .addComponent(updateFee))
            .addGroup(layout.createSequentialGroup()
                .addComponent(enterFee,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(updateFee))
        )
    )
);

```

```

        .addGroup(layout.createSequentialGroup())
            .addComponent(text1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(Menuue,
javax.swing.GroupLayout.PREFERRED_SIZE, 175,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
            .addComponent(addPlace)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(removePlace,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(changeFee,
javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(changePeriod,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(message,
javax.swing.GroupLayout.PREFERRED_SIZE, 441,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 457,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(18, Short.MAX_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())
            .addGap(51, 51, 51)
            .addComponent(enterPeriod,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(354, Short.MAX_VALUE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())
            .addGap(208, 208, 208)
            .addComponent(updatePeriod)
            .addContainerGap(211, Short.MAX_VALUE)))
);
layout.setVerticalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLayeredPanel,
            javax.swing.GroupLayout.PREFERRED_SIZE, 306,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jSeparator1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 10,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(11, 11, 11)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(Menuue,
                javax.swing.GroupLayout.PREFERRED_SIZE, 30,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(text1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 30,
                javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(removePlace,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(changeFee,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(addPlace,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(changePeriod,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(36, 36, 36)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(enterFee,
                javax.swing.GroupLayout.PREFERRED_SIZE, 30,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(updateFee))
            .addGap(26, 26, 26)
            .addComponent(message,
                javax.swing.GroupLayout.PREFERRED_SIZE, 53,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(57, Short.MAX_VALUE))

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addContainerGap(468, Short.MAX_VALUE)
    .addComponent(enterPeriod,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(127, 127, 127)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addContainerGap(470, Short.MAX_VALUE)
    .addComponent(updatePeriod)
    .addGap(126, 126, 126)))

);

pack();
} // </editor-fold> // GEN-END: initComponents

Report report = new Report();

Employee_Invoker invoker = new Employee_Invoker();

static Workshops ComputerGraphics = new Workshops(1, "Computer
Graphics", 200, 10, 21);
static Workshops ComputerSecurity = new Workshops(2, "Computer
Security", 500, 10, 19);
static Workshops ArtificialIntelligence = new Workshops(3, "Artificial
Intelligence", 1000, 10, 37);
static Workshops SoftwareStudio = new Workshops(4, "PersonalCar", 150, 10,
26);
static Workshops FeedbackSystems = new Workshops(5, "Feedback
Systems", 450, 10, 17);

private void BackActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_BackActionPerformed
    close();
    new EmployeeIDPage().setVisible(true);
} // GEN-LAST:event_BackActionPerformed

private void ManageWorkshopActionPerformed(java.awt.event.ActionEvent
evt) { // GEN-FIRST:event_ManageWorkshopActionPerformed

```

```

        addPlace.setVisible(true);
        removePlace.setVisible(true);
        changeFee.setVisible(true);
        text1.setVisible(true);
        Menue.setVisible(true);
        changePeriod.setVisible(true);

    } //GEN-LAST:event_ManageWorkshopActionPerformed

    private void DisplayReportsActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_DisplayReportsActionPerformed
        new Report().setVisible(true);    } //GEN-
    LAST:event_DisplayReportsActionPerformed

    private void HomeActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_HomeActionPerformed
        close();
        new Home().setVisible(true);
    } //GEN-LAST:event_HomeActionPerformed

    private void RemoveSeatsActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_RemoveSeatsActionPerformed

        updateFee.setVisible(false);
        enterFee.setVisible(false);

        Employee_Commands remove;

        if (Menue.getSelectedIndex() == 0)
        {
            remove = new RemovePlace (ComputerGraphics);
            invoker.execute_Employee_Commands(remove); // remove seats
            Report.ComputerGraphics.decrementCount();

            message.setText("Computer Graphics places available: "+
            ComputerGraphics.Count);
        }

        else if (Menue.getSelectedIndex() == 1)
        {

```

```

        remove = new RemovePlace (ComputerSecurity);
        invoker.execute_Employee_Commands(remove); // remove seats
        Report.ComputerSecurity.decrementCount();

        message.setText("Computer Security places available: "+
ComputerSecurity.Count);

    }

    else if (Menue.getSelectedIndex()==2)
    {
        remove = new RemovePlace (ArtificialIntelligence);
        invoker.execute_Employee_Commands(remove); // remove seats
        Report.ArtificialIntelligence.decrementCount();

        message.setText("Artificial Intelligence places available: "+
ArtificialIntelligence.Count);
    }

    else if (Menue.getSelectedIndex()==3)
    {

        remove = new RemovePlace (SoftwareStudio);
        invoker.execute_Employee_Commands(remove); // remove seats
        Report.SoftwareStudio.decrementCount();

        message.setText("Software Studio places available: "+
SoftwareStudio.Count);
    }

    else if (Menue.getSelectedIndex()==4)
    {

        remove = new RemovePlace (FeedbackSystems);
        invoker.execute_Employee_Commands(remove); // remove seats
        Report.FeedbackSystems.decrementCount();

        message.setText("Feedback Systems places available: "+
FeedbackSystems.Count);
    }

    }

}

} //GEN-LAST:event_RemoveSeatsActionPerformed

```

```

    private void AddSeatsActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_AddSeatsActionPerformed

        updateFee.setVisible(false);
        enterFee.setVisible(false);

        Employee_Commands add;
        if (Menue.getSelectedIndex() == 0)
        {
            // to show how command pattern works
            add = new AddPlace (ComputerGraphics);
            invoker.execute_Employee_Commands(add); // add seats

            // this part for the observer pattern to notify others with
the change
            Report.ComputerGraphics.incrementCount();
            message.setText("Computer Graphics places available: "+
ComputerGraphics.Count);
        }
        else if (Menue.getSelectedIndex() == 1)
        {

            // to show how command pattern works
            add = new AddPlace (ComputerSecurity);
            invoker.execute_Employee_Commands(add); // add seats

            // extra for command but needed to notify others with the
change
            Report.ComputerSecurity.incrementCount();
            message.setText("Computer Security places available: "+
ComputerSecurity.Count);
        }

        else if (Menue.getSelectedIndex() == 2)
        {

            // to show how command pattern works
            add = new AddPlace (ArtificialIntelligence);
            invoker.execute_Employee_Commands(add); // add seats

```



```

        // extra for command but needed to notify others with the
change
        Report.ArtificialIntelligence.incrementCount();
        message.setText("Artificial Intelligence places available: "+
ArtificialIntelligence.Count);

    }

    else if (Menu.getSelectedIndex()==3)
    {

        // to show how command pattern works
        add = new AddPlace (SoftwareStudio);
        invoker.execute_Employee_Commands(add); // add seats

        // extra for command but needed to notify others with the
change
        Report.SoftwareStudio.incrementCount();
        message.setText("Software Studio places available: "+
SoftwareStudio.Count);
    }

    else if (Menu.getSelectedIndex()==4)
    {

        // to show how command pattern works
        add = new AddPlace (FeedbackSystems);
        invoker.execute_Employee_Commands(add); // add seats

        // extra for command but needed to notify others with the
change
        Report.FeedbackSystems.incrementCount();

        message.setText("Feedback Systems places available: "+
FeedbackSystems.Count);
    }

} //GEN-LAST:event_AddSeatsActionPerformed

private void changeFeeHomeActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_changeFeeHomeActionPerformed

    updateFee.setVisible(true);
    enterFee.setVisible(true);

    updatePeriod.setVisible(false);
    enterPeriod.setVisible(false);

```

```

    } //GEN-LAST:event_changeFeeHomeActionPerformed

    private void MenueActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_MenueActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_MenueActionPerformed

    private void enterFeeActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_enterFeeActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_enterFeeActionPerformed


    private void updateFeeActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_updateFeeActionPerformed

        Employee_Commands chgFee;

        if (Menue.getSelectedIndex() == 0)
        {
            if (!(checkvalidInt(enterFee.getText()))) {

                ErrorMessage EM = new ErrorMessage("Only enter integers for
the fee");
                EM.setVisible(true);

            } else {
                // for command pattern
                chgFee = new changeFee(ComputerGraphics,
Integer.parseInt(enterFee.getText()));
                invoker.execute_Employee_Commands(chgFee); // change fee

                // this part for observer pattern

                Report.ComputerGraphics.setFee(Integer.parseInt(enterFee.getText()));

                message.setText("The new fee of Computer Graphics workshop: "+
ComputerGraphics.getFee());
            }
        }

        else if (Menue.getSelectedIndex() == 1)
        {
            if (!(checkvalidInt(enterFee.getText()))) {

```

```

        ErrorMessage EM = new ErrorMessage("Only enter integers for
the fee");
        EM.setVisible(true);

        } else {
            chgFee = new changeFee(ComputerSecurity,
Integer.parseInt(enterFee.getText()));
            invoker.execute_Employee_Commands(chgFee); // change fee

Report.ComputerSecurity.setFee(Integer.parseInt(enterFee.getText()));

            message.setText("The new fee of Computer Security workshop: "+
ComputerSecurity.getFee());
        }
    }

    else if (Menue.getSelectedIndex()==2)
    {
        if (!(checkvalidInt(enterFee.getText())) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the fee");
            EM.setVisible(true);

        } else {
            chgFee = new changeFee(ArtificialIntelligence,
Integer.parseInt(enterFee.getText()));
            invoker.execute_Employee_Commands(chgFee); // change fee

Report.ArtificialIntelligence.setFee(Integer.parseInt(enterFee.getText()));

            message.setText("The new fee of Artificial Intelligence
workshop: "+ ArtificialIntelligence.getFee());
        }
    }

    else if (Menue.getSelectedIndex()==3)
    {
        if (!(checkvalidInt(enterFee.getText())) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the fee");
            EM.setVisible(true);

        } else {
            chgFee = new changeFee(SoftwareStudio,
Integer.parseInt(enterFee.getText()));

```

```

        invoker.execute_Employee_Commands(chgFee); // change fee

Report.SoftwareStudio.setFee(Integer.parseInt(enterFee.getText()));

        message.setText("The new fee of Software Studio workshop: "+
SoftwareStudio.getFee());
    }

}

    else if (Menue.getSelectedIndex()==4)
    {
        if (!(checkvalidInt(enterFee.getText()))){

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the fee");

            EM.setVisible(true);

        } else {
            chgFee = new changeFee(FeedbackSystems,
Integer.parseInt(enterFee.getText()));
            invoker.execute_Employee_Commands(chgFee); // change fee

Report.FeedbackSystems.setFee(Integer.parseInt(enterFee.getText()));

            message.setText("The new fee of Feedback Systems workshop: "+
FeedbackSystems.getFee());
        }

    }

}

}

//GEN-LAST:event_updateFeeActionPerformed

private void changePeriodHomeActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_changePeriodHomeActionPerformed

    updateFee.setVisible(false);
    enterFee.setVisible(false);
    updatePeriod.setVisible(true);
    enterPeriod.setVisible(true);

}

//GEN-LAST:event_changePeriodHomeActionPerformed

private void enterPeriodActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_enterPeriodActionPerformed
    // TODO add your handling code here:
}

//GEN-LAST:event_enterPeriodActionPerformed

```

```

private void updatePeriodActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_updatePeriodActionPerformed

    Employee_Commands chgPeriod;

    if (Menue.getSelectedIndex() == 0)
    {
        if (!(checkvalidInt(enterPeriod.getText()))) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the period");
            EM.setVisible(true);

        } else {
            chgPeriod = new changeHours(ComputerGraphics,
Integer.parseInt(enterPeriod.getText()));
            invoker.execute_Employee_Commands(chgPeriod);

Report.ComputerGraphics.setPeriod(Integer.parseInt(enterPeriod.getText()));

            message.setText("The new period of Computer Graphics workshop:
"+ ComputerGraphics.getPeriod());
        }

    }

    else if (Menue.getSelectedIndex()==1)
    {
        if (!(checkvalidInt(enterPeriod.getText()))) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the period");
            EM.setVisible(true);

        } else {
            chgPeriod = new changeHours(ComputerSecurity,
Integer.parseInt(enterPeriod.getText()));
            invoker.execute_Employee_Commands(chgPeriod);

Report.ComputerSecurity.setPeriod(Integer.parseInt(enterPeriod.getText()));

            message.setText("The new period of Computer Security workshop:
"+ ComputerSecurity.getPeriod());
        }
    }
}

```

```

    }

    else if (Menue.getSelectedIndex()==2)
    {
        if (!(checkvalidInt(enterPeriod.getText())) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the period");
            EM.setVisible(true);

        } else {
            chgPeriod = new changeHours(ArtificialIntelligence,
Integer.parseInt(enterPeriod.getText()));
            invoker.execute_Employee_Commands(chgPeriod);

Report.ArtificialIntelligence.setPeriod(Integer.parseInt(enterPeriod.getTex
t()));

            message.setText("The new period of Artificial Intelligence
workshop: "+ ArtificialIntelligence.getPeriod());
        }
    }

    else if (Menue.getSelectedIndex()==3)
    {
        if (!(checkvalidInt(enterPeriod.getText())) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the period");
            EM.setVisible(true);

        } else {
            chgPeriod = new changeHours(SoftwareStudio,
Integer.parseInt(enterPeriod.getText()));
            invoker.execute_Employee_Commands(chgPeriod);

Report.SoftwreStudio.setPeriod(Integer.parseInt(enterPeriod.getText()));

            message.setText("The new period of Software Studio workshop: "+
SoftwareStudio.getPeriod());
        }
    }

    else if (Menue.getSelectedIndex()==4)
    {
        if (!(checkvalidInt(enterPeriod.getText())) {

            ErrorMessage EM = new ErrorMessage("Only enter integers for
the period");
            EM.setVisible(true);

```

```

        } else {
            chgPeriod = new changeHours(FeedbackSystems,
Integer.parseInt(enterPeriod.getText()));
            invoker.execute_Employee_Commands(chgPeriod);

Report.FeedbackSystems.setPeriod(Integer.parseInt(enterPeriod.getText()));

            message.setText("The new period of Feedback Systems workshop:
"+ FeedbackSystems.getPeriod());
        }
    }

}

}

//GEN-LAST:event_updatePeriodActionPerformed

static boolean checkvalidInt(String number) {
    try {
        Integer.parseInt(number);
    } catch (NumberFormatException e) {
        return false;
    }
    return true;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton Back;
private javax.swing.JButton Back1;
private javax.swing.JButton DisplayReports;
private javax.swing.JButton Manage;
private javax.swing.JComboBox Menuue;
private javax.swing.JButton addPlace;
private javax.swing.JButton changeFee;
private javax.swing.JButton changePeriod;
private javax.swing.JTextField enterFee;
private javax.swing.JTextField enterPeriod;
private javax.swing.JLabel jLabel2;
private javax.swing.JLayeredPane jLayeredPanel;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JLabel message;
private javax.swing.JButton removePlace;
private javax.swing.JLabel text1;
private javax.swing.JButton updateFee;
private javax.swing.JButton updatePeriod;
// End of variables declaration//GEN-END:variables
}

```

Manager_Commands.java

```
package Esraa_SP_Project;

public interface Manager_Commands {

    public void execute();

}
```

Manager_Invoker.java

```
package Esraa_SP_Project;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author esraaaljahdali
 */
public class Manager_Invoker {
    private List<Manager_Commands> M = new ArrayList<Manager_Commands>();

    void execute_Manager_Commands(Manager_Commands command) {

        this.M.add(command); // optional
        command.execute();

    }
}
```

Memento.java

```
package Esraa_SP_Project;
```



```

public class Memento {

    private Participant participant;

    public Memento(Participant p)
    {
        participant = p;
    }

    public Participant getState()
    {
        return participant;
    }
}

```

Originator.java

```

package Esraa_SP_Project;

public class Originator
{

    private Participant participant;

    //Setting object
    public void setParticipant(Participant p)
    {
        participant = p;
    }

    public Memento saveMemento()
    {
        return new Memento(participant);
    }

    public void restoreMemento(Memento memento)
    {
        participant = memento.getState();
    }

    public Participant getParticipant()

```

```

    {
        return participant;
    }

}

```

Participant.java

```

package Esraa_SP_Project;

public class Participant {
    int ID;
    String Fname;
    String Lname;
    String Type;
    String phone;
    String address;
    String email;
    String city;
    String state;
    String country;
    String zip;
    int countryINDEX;

    public Participant(int id,String fname,String lname,String type, String
Phone, String Address, String Email, String City, String State, String
Country, String Zip, int countryIndex )
    {
        ID = id;
        Fname = fname;
        Lname = lname;
        Type = type;
        phone = Phone;
        address = Address;
        email = Email;
        city = City;
        state = State;
        country = Country;
        zip = Zip;
        countryINDEX = countryIndex;
    }
}

```

```

    }

    public void setID(int id)
    {
        ID = id;
    }
    public int getID()
    {
        return ID;
    }

    public void setFname(String fname)
    {
        Fname = fname;
    }
    public String getFame()
    {
        return Fname;
    }

    public void setLname(String lname)
    {
        Lname = lname;
    }
    public String getLame()
    {
        return Lname;
    }

    public void setWorkshopType(String type)
    {
        Type = type;
    }

    public String getWorkshopType()
    {
        return Type;
    }

    public void setPhone(String Phone)
    {
        phone = Phone;
    }
    public String getPhone()
    {
        String phoneOutput = "(" + phone.substring( 0,3 ) + ") " +
phone.substring( 3,6 ) + "-" + phone.substring( 6,10 );

```

```
        return phoneOutput;
    }

    public void setAddress(String Address)
    {
        address = Address;
    }
    public String getAddress()
    {
        return address;
    }


    public void setEmail(String Email)
    {
        email = Email;
    }
    public String getEmail()
    {
        return email;
    }


    public void setCity(String City)
    {
        city = City;
    }
    public String getCity()
    {
        return city;
    }


    public void setState(String State)
    {
        state = State;
    }
    public String getState()
    {
        return state;
    }


    public void setCountry(String Country)
    {
        country = Country;
    }
    public String getCountry()
    {
        return country;
    }
}
```

```

public void setZip(String Zip)
{
    zip = Zip;
}
public String getZip()
{
    return zip;
}

public void getCountryIndex(int index)
{
    countryINDEX = index;
}
public int getCountryIndex()
{
    return countryINDEX;
}

}

```

ParticipantArrayList.java

```

package Esraa_SP_Project;

import java.util.ArrayList;

public class ParticipantArrayList {

    ArrayList as = new ArrayList();

    public void addParticipant(Participant p){
        as.add(p);
    }

    public void removeParticipant(Participant p)
    {
        as.remove(p);
    }

}

```

```

    public int getSize()
    {
        return as.size();
    }

    public Participant getParticipant(int i)
    {
        return (Participant)as.get(i);
    }
}

```

ParticipantPage.java

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;

/**
 *
 * @author Esraa Aljahdali
 */
public class ParticipantPage extends javax.swing.JFrame {

    /**
     * Creates new form ParticipantPage
     */
    public ParticipantPage() {
        initComponents();
    }

    public void close()
    {

```

```

        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }
    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> // GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel = new javax.swing.JLayeredPane();
        btnReservation = new javax.swing.JButton();
        btnCancel = new javax.swing.JButton();
        btnback1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        jLayeredPanel.setBackground(new java.awt.Color(204, 255, 255));

        btnReservation.setBackground(new java.awt.Color(220, 230, 230));
        btnReservation.setFont(new java.awt.Font("Tahoma", 1, 13)); //
NOI18N
        btnReservation.setText("New Registration");
        btnReservation.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnReservationActionPerformed(evt);
            }
        });
        jLayeredPanel.add(btnReservation);
        btnReservation.setBounds(150, 190, 170, 60);

        btnCancel.setBackground(new java.awt.Color(250, 209, 223));
        btnCancel.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        btnCancel.setForeground(new java.awt.Color(51, 0, 0));
        btnCancel.setText("Cancel Registration");
        btnCancel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCancelActionPerformed(evt);
            }
        });
    }

```

```

    });
    jLayeredPanel1.add(btnCancel);
    btnCancel.setBounds(150, 260, 170, 60);

    btnback1.setBackground(new java.awt.Color(244, 242, 242));
    btnback1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
    btnback1.setText("Home");
    btnback1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnback1ActionPerformed(evt);
        }
    });
    jLayeredPanel1.add(btnback1);
    btnback1.setBounds(150, 330, 170, 50);

    jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/CMTool.gif"
))); // NOI18N
    jLayeredPanel1.add(jLabel1);
    jLabel1.setBounds(0, 0, 460, 330);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 464, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLayeredPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 425,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void btnReservationActionPerformed(java.awt.event.ActionEvent
evt) { // GEN-FIRST:event_btnReservationActionPerformed
    close();
    new Registration().setVisible(true);
} // GEN-LAST:event_btnReservationActionPerformed

```



```

        private void btnCancelActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_btnCancelActionPerformed
        close();
        new CancelRegistration().setVisible(true);
    } //GEN-LAST:event_btnCancelActionPerformed

    private void btnbacklActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_btnbacklActionPerformed
        close();
        new Home().setVisible(true);
    } //GEN-LAST:event_btnbacklActionPerformed

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JButton btnCancel;
    private javax.swing.JButton btnReservation;
    private javax.swing.JButton btnbackl;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLayeredPane jLayeredPanel;
    // End of variables declaration //GEN-END:variables
}

```

Workshops.java

```

package Esraa_SP_Project;

import java.util.Observable;

public class Workshops extends Observable{
    int ID;
    String Type;
    int Fee;
    int Period;
    int Count;

    public Workshops(int id, String type, int fee, int count, int period)
    {
        ID = id;
        Type = type;
        Fee = fee;
        Count = count;
        Period = period;
    }
}

```

```
public void setWorkshopID(int id)
{
    ID=id;
}
public int getWorkshopID()
{
    return ID;
}

public void setType(String type)
{
    this.Type = type;
}

public String getType()
{
    return Type;
}

public void setPeriod(int period)
{
    Period = period;
}
public int getPeriod()
{
    return Period;
}

public void setFee(int fee)
{
    Fee = fee;
}

public int getFee()
{
    return Fee;
}

public void setCount(int count)
{
    Count=count;
}

// to count the number of places in workshops
public int getCount()
{
    return Count;
}
```

```

public boolean decrementCount ()
{
    if (this.Count==0)
    {
        return false;
    }
    else
    {
        Count--;
    }
    return true;
}

public void incrementCount ()
{
    Count++;
}

public void Report ()
{
    if (this.getWorkshopID()==1)
    {
        setChanged();

        // notify oservers that CG is chosing to get increment or
decrement || feeChange || periodchange
        notifyObservers("Computer Graphics");
    }

    else if (this.getWorkshopID()==2)
    {
        setChanged();
        notifyObservers("Computer Security");
    }

    else if (this.getWorkshopID()==3)
    {
        setChanged();
        notifyObservers("Artificial Intelligence");
    }

    else if (this.getWorkshopID()==4)
    {
        setChanged();
        notifyObservers("Software Studio");
    }

    else if (this.getWorkshopID()==5)

```

```

        {
            setChanged();
            notifyObservers("Feedback Systems");
        }
    }

    public int totalWorkshopPlaces()
    {
        ComputerGraphics g = new ComputerGraphics();
        ComputerSecurity s = new ComputerSecurity();
        ArtificialIntelligence l = new ArtificialIntelligence();
        SoftwareStudio o = new SoftwareStudio();
        FeedbackSystems f = new FeedbackSystems();

        int total = g.Count + s.Count + l.Count + o.Count + f.Count;

        return total;
    }
}

```

Report.java

```

package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.util.Observable;
import java.util.Observer;

public class Report extends javax.swing.JFrame implements Observer{

    public Report() {
        initComponents();
        setTitle("Reports");
    }

    //-----
    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent

```

```

);
}

//-----
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        Back = new javax.swing.JButton();
        WorkshopMenue = new javax.swing.JComboBox();
        txtreport = new javax.swing.JLabel();
        DisplayReportprint = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(0, 0, 153));
        jLabel1.setText("Choose a Workshop to Display a report :");

        Back.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        Back.setText("Home");
        Back.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BackActionPerformed(evt);
            }
        });

        WorkshopMenue.setFont(new java.awt.Font("Lucida Grande", 1, 13));
// NOI18N
        WorkshopMenue.setForeground(new java.awt.Color(0, 0, 204));
        WorkshopMenue.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Computer Graphics", "Computer Security", "Software Studio",
"Artificial Intelligence", "Feedback Systems" }));
        WorkshopMenue.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                WorkshopMenueActionPerformed(evt);
            }
        });

        txtreport.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        txtreport.setForeground(new java.awt.Color(0, 51, 102));

        DisplayReportprint.setBackground(new java.awt.Color(249, 234,
205));
        DisplayReportprint.setFont(new java.awt.Font("Tahoma", 1, 13)); //

```

```

NOI18N
    DisplayReportprint.setText("Display Report");
    DisplayReportprint.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            DisplayReportprintActionPerformed(evt);
        }
    });

    jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/reportIcon.
gif"))); // NOI18N

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap()
            .addComponent(txtreport,
javax.swing.GroupLayout.PREFERRED_SIZE, 370,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(57, 57, 57))
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGap()
                        .addComponent(jLabel1))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(153, 153, 153)
                        .addComponent(WorkshopMenue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap()
                    .addComponent(jLabel2))
                .addGroup(layout.createSequentialGroup()
                    .addGap(177, 177, 177)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(DisplayReportprint,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

```

```

        .addComponent(Back,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
        .addGap(7, 7, 7))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel2)
            .addGap(18, 18, 18)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(WorkshopMenuue,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 52,
Short.MAX_VALUE)
            .addComponent(DisplayReportprint)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(Back, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(txtreport,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(14, 14, 14))
        );

    pack();
} // </editor-fold> // GEN-END: initComponents

```

```

    static Workshops ComputerGraphics = new Workshops(1, "Computer
Graphics", 200, 10, 21);
    static Workshops ComputerSecurity = new Workshops(2, "Computer
Security", 500, 10, 19);
    static Workshops ArtificialIntelligence = new Workshops(3, "Artificial
Intelligence", 1000, 10, 37);
    static Workshops SoftwareStudio = new Workshops(4, "Software Studio",
150, 10, 26);
    static Workshops FeedbackSystems = new Workshops(5, "Artificial
Intelligence", 450, 10, 17);

```

```

        private void BackActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_BackActionPerformed
        close();
        new Home().setVisible(true);
    }
    //GEN-LAST:event_BackActionPerformed

// when call notifyobserver("") <- Object message
public void update(Observable workshop, Object message)
{
    Workshops WS = (Workshops)workshop;
    String msg = (String)message;
    //String print="";

    if(msg.equalsIgnoreCase("Computer Graphics"))
    {
        WS = ComputerGraphics;

        String txt = "The number of places available: ";
        String txt2 = "" + WS.Count;
        String txt3 = "The fee of the workshop: ";
        String txt4 = "" + WS.getFee();
        String txt5 = "The period of the workshop: ";
        String txt6 = "" + WS.getPeriod();

        close();
        // display the report in a new message box
        MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5,
txt6, msg);
        mb.setVisible(true);

    }

    else if (msg.equalsIgnoreCase("Computer Security"))
    {
        WS = ComputerSecurity;

        String txt = "The number of places available: ";
        String txt2 = "" + WS.Count;
        String txt3 = "The fee of the workshop: ";
        String txt4 = "" + WS.getFee();
        String txt5 = "The period of the workshop: ";
        String txt6 = "" + WS.getPeriod();

        close();
    }
}

```



```

        MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5,
txt6, msg);
        mb.setVisible(true);

    }

    else if (msg.equalsIgnoreCase("Artificial Intelligence"))
    {
        WS = ArtificialIntelligence;

        String txt = "The number of places available: ";
        String txt2 = "" +WS.Count;
        String txt3 = "The fee of the workshop: ";
        String txt4 = "" + WS.getFee();
        String txt5 = "The period of the workshop: ";
        String txt6 = "" + WS.getPeriod();

        close();
        MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5,
txt6, msg);
        mb.setVisible(true);

    }

    else if (msg.equalsIgnoreCase("Software Studio"))
    {
        WS = SoftwareStudio;
        String txt = "The number of places available: ";
        String txt2 = "" +WS.Count;
        String txt3 = "The fee of the workshop: ";
        String txt4 = "" + WS.getFee();
        String txt5 = "The period of the workshop: ";
        String txt6 = "" + WS.getPeriod();

        close();
        MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5,
txt6, msg);
        mb.setVisible(true);
    }

    else if (msg.equalsIgnoreCase("Feedback Systems"))
    {
        WS = FeedbackSystems;

        String txt = "The number of places available: ";
        String txt2 = "" +WS.Count;
        String txt3 = "The fee of the workshop: ";

```

```

        String txt4 = "" + WS.getFee();
        String txt5 = "The period of the workshop: ";
        String txt6 = "" + WS.getPeriod();

        close();
        MessageBox2 mb = new MessageBox2(txt, txt2, txt3, txt4, txt5,
txt6, msg);
        mb.setVisible(true);
    }
    //To change body of generated methods, choose Tools | Templates.
}
private void WorkshopMenuActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_WorkshopMenuActionPerformed

    //GEN-LAST:event_WorkshopMenuActionPerformed

    private void
DisplayReportprintActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_DisplayReportprintActionPerformed
        dispose();

        Report report = new Report();

        if(WorkshopMenu.getSelectedIndex() == 0)
        {
            //txtreport.setText("Computer Graphics");
            // ComputerGraphics.setType("Computer Graphics");
            ComputerGraphics.addObserver(report);
            ComputerGraphics.Report();
        }

        else if(WorkshopMenu.getSelectedIndex() == 1)
        {
            ComputerSecurity.setType("Computer Security");
            ComputerSecurity.addObserver(report);
            ComputerSecurity.Report();
        }

        else if(WorkshopMenu.getSelectedIndex() == 2)
        {
            SoftwareStudio.setType("Software Studio");
            SoftwareStudio.addObserver(report);
            SoftwareStudio.Report();
        }

        else if(WorkshopMenu.getSelectedIndex() == 3)
        {
            ArtificialIntelligence.setType("Artificial Intelligence");

```

```

        ArtificialIntelligence.addObserver(report);
        ArtificialIntelligence.Report();
    }

    else if(WorkshopMenu.getSelectedIndex() == 4)
    {
        FeedbackSystems.setType("Feedback Systems");
        FeedbackSystems.addObserver(report);
        FeedbackSystems.Report();
    }
} //GEN-LAST:event_DisplayReportprintActionPerformed

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton Back;
private javax.swing.JButton DisplayReportprint;
private javax.swing.JComboBox WorkshopMenu;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel txtreport;
// End of variables declaration//GEN-END:variables
}

```

ErrorMessage.java

```

package Esraa_SP_Project;

import java.awt.Color;
import java.awt.Font;
import javax.swing.UIManager;
import javax.swing.plaf.ColorUIResource;

/**
 *
 * @author esraaaljahdali
 */
public class ErrorMessage extends javax.swing.JFrame {

    /**
     * Creates new form ErrorMessage
     */
    public ErrorMessage(String s) {
        initComponents();
        txt.setText(s);
    }

```

```

        setTitle("Error Message");
        UIManager.put("InternalFrame.activeTitleForeground", new
ColorUIResource(Color.WHITE));
        // UIManager.put("InternalFrame.titleFont", new Font("Dialog",
Font.PLAIN, 11));
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> // GEN-
BEGIN: initComponents
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        txt = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jButton1.setText("OK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/errorIcon.p
ng"))); // NOI18N

        txt.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        txt.setForeground(new java.awt.Color(172, 44, 44));
        txt.setText("jLabel1");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

layout.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

ING)

        .addGroup(layout.createSequentialGroup()
            .addGap(43, 43, 43)
            .addComponent(jLabel2)
            .addGap(18, 18, 18)
            .addComponent(txt,
javax.swing.GroupLayout.PREFERRED_SIZE, 472,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(246, 246, 246)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(20, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap(53, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAI
LING)

            .addComponent(jLabel2)
            .addComponent(txt,
javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(34, 34, 34)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(32, 32, 32))
    );

    pack();
} // </editor-fold> //GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton1ActionPerformed
    dispose();
} //GEN-LAST:event_jButton1ActionPerformed

/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton jButton1;

```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel txt;
// End of variables declaration//GEN-END:variables
}

```

FinalMessage.java

```

package Esraa_SP_Project;

import java.awt.Toolkit;
import java.awt.event.WindowEvent;

/**
 *
 * @author Esraa Aljahdali
 */
public class FinalMessage extends javax.swing.JFrame {

    /**
     * Creates new form FinalMessage
     */
    public FinalMessage() {
        initComponents();
        print();
    }

    public void print()
    {
        MessageLabel.setText("Payment and Registration was successful");
    }
    /**
     * This method is called from within the constructor to initialize the
    form.
     * WARNING: Do NOT modify this code. The content of this method is
    always
     * regenerated by the Form Editor.
     */
}

```



```

javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(106, 106, 106)
        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 105,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(72, 72, 72))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addComponent(MessageLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(56, 56, 56))))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGap(123, 123, 123)
            .addComponent(MessageLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(70, 70, 70)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASE
LINE)
            .addComponent(backbtn,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(68, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

public void close()
{
    WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

    Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
}

private void backbtnActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_backbtnActionPerformed
    close();
    new Home().setVisible(true);
}

```



```

        // TODO add your handling code here:
    } //GEN-LAST:event_backbtnActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton1ActionPerformed
        close();    } //GEN-LAST:event_jButton1ActionPerformed

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JLabel MessageLabel;
    private javax.swing.JButton backbtn;
    private javax.swing.JButton jButton1;
    // End of variables declaration //GEN-END:variables
}

```

Main1.com

```

package Esraa_SP_Project;

/**
 *
 * @author esraaaljahdali
 */
public class Main1 {

    /**
     * @param args the command line arguments
     */

    public static void main(String args[]) {

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Main1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(Main1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Main1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Main1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    //java.awt.EventQueue.invokeLater(new Runnable() {
    //    public void run() {

        new Home().setVisible(true);
    //    }
    // });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
// End of variables declaration//GEN-END:variables
}

```

MessageBox.java

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Esraa_SP_Project;

import java.awt.Toolkit;

```

```

import java.awt.event.WindowEvent;

/**
 *
 * @author Esraa Aljahdali
 */
public class MessageBox extends javax.swing.JFrame {

    /**
     * Creates new form MessageBox
     * @param s1
     */

    Participant p;

    public MessageBox(String s1, String s2, Participant p) {
        initComponents();
        txtMessage.setText(s1);
        txtMessage1.setText(s2);
        this.p = p;
    }

    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }
    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLayeredPanel = new javax.swing.JLayeredPane();
        txtMessage = new javax.swing.JLabel();
        back = new javax.swing.JButton();
        confirm = new javax.swing.JButton();
        txtMessage1 = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

txtMessage.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLayeredPanel.add(txtMessage);
txtMessage.setBounds(90, 90, 330, 50);

back.setBackground(new java.awt.Color(232, 232, 249));
back.setText("Back");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
jLayeredPanel.add(back);
back.setBounds(130, 190, 80, 40);

confirm.setBackground(new java.awt.Color(227, 250, 227));
confirm.setText("Confirm");
confirm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        confirmActionPerformed(evt);
    }
});
jLayeredPanel.add(confirm);
confirm.setBounds(290, 190, 80, 40);

txtMessage1.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
jLayeredPanel.add(txtMessage1);
txtMessage1.setBounds(90, 40, 330, 50);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLayeredPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 504,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLayeredPanel,
javax.swing.GroupLayout.DEFAULT_SIZE, 252, Short.MAX_VALUE)
        .addContainerGap())

```

```

);

pack();
} // </editor-fold> // GEN-END: initComponents

private void confirmActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_confirmActionPerformed

    ManagerPage_participant.as.addParticipant(p);

    if (Registration.select.equalsIgnoreCase("Computer Graphics"))
    {

        if (Report.ComputerGraphics.decrementCount())
        {
            close();
            new FinalMessage().setVisible(true);
        }
    }

    else if (Registration.select.equalsIgnoreCase("Artificial
Intelligence"))
    {
        if (Report.ArtificialIntelligence.decrementCount())
        {
            close();
            new FinalMessage().setVisible(true);
        }
    }

    else if (Registration.select.equalsIgnoreCase("Computer Security"))
    {
        if (Report.ComputerSecurity.decrementCount())
        {
            close();
            new FinalMessage().setVisible(true);
        }
    }

    else if (Registration.select.equalsIgnoreCase("Software Studio"))
    {
        if (!Report.SoftwareStudio.decrementCount())
        {
            close();
            new FinalMessage().setVisible(true);
        }
    }

```

```

    }

    else if (Registration.select.equalsIgnoreCase("Feedback Systems"))
    {
        if (!Report.FeedbackSystems.decrementCount())
        {
            close();
            new FinalMessage().setVisible(true);
        }
    }

}

} //GEN-LAST:event_confirmActionPerformed

private void backActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_backActionPerformed
    close();
    new Registration().setVisible(true);
} //GEN-LAST:event_backActionPerformed

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton back;
private javax.swing.JButton confirm;
private javax.swing.JLayeredPane jLayeredPanel;
private javax.swing.JLabel txtMessage;
private javax.swing.JLabel txtMessage1;
// End of variables declaration //GEN-END:variables
}

```

MessageBox2.java

```

package Esraa_SP_Project;

import java.awt.event.WindowEvent;
import java.awt.Toolkit;

/**
 *
 * @author esraaaljhdali
 */
public class MessageBox2 extends javax.swing.JFrame {

    /**

```

```

    * Creates new form MessageBox2
    */
    public MessageBox2(String s1, String s2, String s3, String s4, String
s5, String s6, String msg) {
        initComponents();
        txt.setText(s1);
        txt1.setText(s2);
        txt2.setText(s3);
        txt3.setText(s4);
        txt4.setText(s5);
        txt5.setText(s6);
        setTitle(msg + " Report");
        //Back.setText(b1);
        //Exit.setText(b2);
    }

    public void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent
);
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN: initComponents
    private void initComponents() {

        Exit = new javax.swing.JButton();
        Back = new javax.swing.JButton();
        txt = new javax.swing.JLabel();
        txt1 = new javax.swing.JLabel();
        txt2 = new javax.swing.JLabel();
        txt3 = new javax.swing.JLabel();
        txt4 = new javax.swing.JLabel();
        txt5 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

Exit.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Exit.setText("Exit");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});

Back.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Back.setText("Back");
Back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BackActionPerformed(evt);
    }
});

txt.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
txt.setForeground(new java.awt.Color(51, 0, 255));
txt.setText("jLabel1");

txt1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt1.setText("jLabel1");

txt2.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
txt2.setForeground(new java.awt.Color(51, 0, 255));
txt2.setText("jLabel1");

txt3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt3.setText("jLabel1");

txt4.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
txt4.setForeground(new java.awt.Color(51, 0, 255));
txt4.setText("jLabel1");

txt5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt5.setText("jLabel1");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAI
LING)

        .addGroup(layout.createSequentialGroup()
            .addContainerGap(107, Short.MAX_VALUE)

```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(txt2,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txt1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txt,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txt3,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txt4,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(txt5,
        javax.swing.GroupLayout.PREFERRED_SIZE, 304,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(50, 50, 50))
.addGroup(layout.createSequentialGroup()
    .addGap(46, 46, 46)
    .addComponent(Exit,
        javax.swing.GroupLayout.PREFERRED_SIZE, 120,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(Back,
        javax.swing.GroupLayout.PREFERRED_SIZE, 119,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(59, 59, 59))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addGap(43, 43, 43)
    .addComponent(txt, javax.swing.GroupLayout.PREFERRED_SIZE,
40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txt1, javax.swing.GroupLayout.PREFERRED_SIZE,
40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txt2, javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txt3, javax.swing.GroupLayout.PREFERRED_SIZE,
40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txt4, javax.swing.GroupLayout.PREFERRED_SIZE,
40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txt5, javax.swing.GroupLayout.PREFERRED_SIZE,
40, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASE
LINE)
    .addComponent(Exit)
    .addComponent(Back))
    .addGap(44, 44, 44))
);

pack();
} // </editor-fold> // GEN-END: initComponents

private void ExitActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_ExitActionPerformed
    close();
} // GEN-LAST: event_ExitActionPerformed

private void BackActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_BackActionPerformed
    dispose();
    new Report().setVisible(true);
} // GEN-LAST: event_BackActionPerformed

// Variables declaration - do not modify // GEN-BEGIN: variables
private javax.swing.JButton Back;
private javax.swing.JButton Exit;
private javax.swing.JLabel txt;
private javax.swing.JLabel txt1;
private javax.swing.JLabel txt2;
private javax.swing.JLabel txt3;
private javax.swing.JLabel txt4;
private javax.swing.JLabel txt5;
// End of variables declaration // GEN-END: variables

```

```
}
```

SuccessMessage.java

```
package Esraa_SP_Project;

/**
 *
 * @author esraaaljahdali
 */
public class SuccessMessage extends javax.swing.JFrame {

    /**
     * Creates new form SuccessMessage
     */
    public SuccessMessage(String s) {
        initComponents();
        txt.setText(s);
    }

    /**
     * This method is called from within the constructor to initialize the
     form.
     * WARNING: Do NOT modify this code. The content of this method is
     always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
    BEGIN: initComponents
        private void initComponents() {

            jButton1 = new javax.swing.JButton();
            jLabel1 = new javax.swing.JLabel();
            txt = new javax.swing.JLabel();

            setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

            jButton1.setText("OK");
            jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    jButton1ActionPerformed(evt);
                }
            });
        }
    }
}
```

```

        jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Esraa_SP_Project/accept-
icon.png"))); // NOI18N

        txt.setFont(new java.awt.Font("Tahoma", 1, 13)); // NOI18N
        txt.setText("jLabel2");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(52, 52, 52)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(txt, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addGroup(layout.createSequentialGroup()
                    .addGap(228, 228, 228)
                    .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(240, 240, 240)
                )
            )
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(51, 51, 51)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel1)
                    .addComponent(txt,
javax.swing.GroupLayout.PREFERRED_SIZE, 39,
javax.swing.GroupLayout.PREFERRED_SIZE)
                )
                .addGap(27, 27, 27)
                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(55, 55, 55)
            )
        );

        pack();
    } // </editor-fold> // GEN-END: initComponents

```

```
        private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton1ActionPerformed
    dispose();
} //GEN-LAST:event_jButton1ActionPerformed


// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel txt;
// End of variables declaration //GEN-END:variables
}
```